PENGEMBANGAN SISTEM INFORMASI PENGGAJIAN PEGAWAI DAN DOKUMENTASI PEKERJAAN BERBASIS WEB (STUDI KASUS: PT. YUGA ELEKTRO TUBAN)

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Raga Saputra Heri Istanto
NIM: 145150400111023

PROGRAM STUDI SISTEM INFORMASI
JURUSAN SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

PENGEMBANGAN SISTEM INFORMASI PENGGAJIAN PEGAWAI DAN DOKUMENTASI PEKERJAAN BERBASIS WEB
(STUDI KASUS: PT. YUGA ELEKTRO TUBAN)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh:
Raga Saputra Heri Istanto
NIM: 145150400111023

Skripsi ini telah diuji dan dinyatakan lulus pada
7 Juni 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Ismiarta Aknuranda, S.T, M.Sc, Ph.D
NIK. 201006 740719 1 001

Dosen Pembimbing II

Moch. Chandra Saputra, S.Kom, M.T, M.Eng.
NIK. 201609 860106 1 001

Mengetahui
Ketua Jurusan Sistem Informasi

Herman Tolle, Dr. Eng., S.T, M.T
NIP: 19740823 200012 1 001

IDENTITAS TIM PENGUJI

Informasi pelaksanaan ujian dan penguji.

JADWAL
Jadwal pelaksanaan ujian

Hari/Tanggal	KAMIS / 07 JUNI 2018
Tempat	FILKOM UB / B1.6
Waktu	08:00 - 09:15

PENGUJI
Majelis penguji ujian skripsi

Welly Purnomo, S.T., M.Kom. (ke I) * ketua majelis
NIK. 81011706110026

Rizal Setya Perdana, S.Kom, M.Kom (ke II)
NIK. 201603 910118 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenarya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata di dalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 7 Juni 2018

Raga Saputra Heri Istanto
NIM: 145150400111023

DAFTAR RIWAYAT HIDUP

Data Pribadi

Nama lengkap	: Raga Saputra Heri Istanto
Tempat, tanggal lahir	: Tuban, 24 Desember 1995
Jenis kelamin	: Laki-Laki
Status	: Belum Kawin
Kewarganegaraan	: Indonesia
Alamat	: Jl. MT.Haryono 2 RT.02 RW.01, No 489, Belakang BCA, Dinoyo, Malang.
No. HP	: 081233069975
Email	: heriistantoo@gmail.com

Latar Belakang Pendidikan

Pendidikan Formal

2014-Sekarang	: S1 Sistem Informasi Fakultas Ilmu Komputer Universitas Brawijaya, IPK = 3.74 (Skala = 4)
2011-2014	: SMAN 2 Tuban
2008-2011	: SMPN 1 Jatirogo
2005-2008	: SDN 1 Jatirogo
2003-2005	: TK Tunas Rimba Jatirogo

Pengalaman Organisasi

1. Asisten Praktikum mata kuliah Dasar Basis Data TA.2017/2018
2. Asisten Praktikum mata kuliah Sistem Basis Data TA.2016/2017
3. Asisten Praktikum mata kuliah Administrasi Basis Data TA.2016/2017
4. Asisten Praktikum mata kuliah Pemrogaman Lanjut TA.2015/2016
5. Pendamping Study Club "Crystal" (mata kuliah Pemrogaman Dasar)
6. Pendamping WOW (Week Of Welcoming) Sistem Informasi
7. Anggota ROBOTIK SMAN 2 Tuban
8. Anggota SISPELIDUP SMAN 2 Tuban (Siswa Pecinta Lingkungan Hidup)
9. Anggota OSIS SMPN 1 Jatirogo

Prestasi

1. Finalis Carbon 1.5 FILKOM
2. Juara Desain Terbaik, Kontes Robot ROV di ITS Surabaya

UCAPAN TERIMAKASIH

Penelitian ini tentunya tidak bisa terselesaikan tanpa dukungan dan bantuan dari orang-orang terdekat, sehingga saya ingin menyampaikan rasa terimakasih kepada:

1. Ibu saya, Supik yang senantiasa memberikan dukungan doa, moril, materil, mengajarkan ketegasan, dan kedisiplinan dalam menyelesaikan penelitian. Serta senantiasa memberikan contoh yang baik untuk menjadi pribadi yang sabar dan pekerja keras. Ayah saya, Eko Heri Istanto yang telah memberikan kesempatan dan kemudahan untuk dapat melakukan penelitian di lokasi studi kasus dan saya yakini juga memberikan dukungan doa dan menjadi pribadi yang bersahabat dengan saya.
2. Bapak Ismiarta Aknuranda, S.T, M.Sc, Ph.D, selaku pembimbing satu yang mengajarakan ketegasan dan kedisiplinan dalam menyelesaikan kewajiban serta telah menjadi pembimbing yang sabar, solutif dan memberikan contoh sebagai pribadi yang profesional dalam memberikan bimbingan kepada mahasiswa.
3. Bapak Mochamad Chandra Saputra, S.Kom, M.Eng, selaku pembimbing dua yang telah menjadi pembimbing yang sabar, solutif, dan memberikan contoh sebagai pribadi yang profesional dalam memberikan bimbingan kepada mahasiswa.
4. Mirna Suryani, dan segenap pegawai PT. Yuga Elektro Tuban yang memberikan kemudahan pada proses penelitian di lokasi studi kasus.
5. Teman-teman Jurusan Sistem Informasi, Fakultas Ilmu Komputer, yang tidak dapat saya sebutkan namanya satu per satu, yang telah memberikan saya dukungan moril, menjadi teman dan sahabat yang baik dan menyenangkan selama masa perkuliahan.

Semoga Allah SWT melimpahkan rahmat berlipat ganda kepada seluruh pihak yang membantu dan berkontribusi pada penelitian ini. Tidak lupa saya sampaikan bahwa penelitian yang telah dilakukan masih banyak memiliki kekurangan, diskusi mengenai penelitian lanjutan, saran, dan kritik yang membangun merupakan beberapa hal yang saya harapkan dapat disampaikan kepada saya.

ABSTRAK

Raga Saputra Heri Istanto, Pengembangan Sistem Informasi Penggajian Pegawai dan Dokumentasi Pekerjaan Berbasis Web (Studi Kasus: PT. Yuga Elektro Tuban)

Pembimbing: Ismiarta Aknuranda dan Mochamad Chandra Saputra

Kurangnya dukungan teknologi pada perusahaan yuga elektro tuban mengakibatkan proses pekerjaan pembuatan laporan keuangan dan pembuatan dokumentasi pekerjaan yang dilakukan oleh petugas administrasi menjadi tidak efisien. Di dalam usaha untuk meningkatkan efisiensi pekerjaan, dibutuhkan dukungan teknologi yang menyediakan fungsi-fungsi untuk mendukung kedua pekerjaan tersebut melalui sebuah sistem informasi sehingga kecepatan kerja petugas administrasi menjadi lebih baik.

Untuk memenuhi kebutuhan dukungan teknologi melalui sebuah sistem informasi, penelitian ini dilakukan untuk mengetahui dapatkah dibangun sebuah prototipe sistem informasi yang mampu mendemonstrasikan fungsi-fungsi untuk mendukung kedua pekerjaan tersebut dengan baik. Menjawab masalah tersebut, dilakukan analisis persyaratan, perancangan, dan implementasi sistem informasi dengan menerapkan pendekatan berorientasi objek. Kemudian, hasil implementasi sistem diuji untuk menentukan apakah sistem dapat berjalan sesuai dengan persyaratan yang telah teridentifikasi.

Kegiatan analisis persyaratan menghasilkan spesifikasi persyaratan sistem informasi dan visualisasi kemampuan sistem dalam bentuk *use case diagram*. Hasil kegiatan perancangan menghasilkan dokumentasi model interaksi objek ke dalam bentuk *sequence diagram,* model objek ke dalam *class diagram,* rancangan basis data, rancangan algoritme, dan rancangan antarmuka pengguna. Hasil implementasi merupakan sebuah prototipe sistem informasi yang mampu mendemonstrasikan fungsi-fungsi yang telah teridentifikasi pada hasil spesifikasi persyaratan sistem informasi. Prototipe sistem informasi selanjutnya diuji berdasarkan persyaratan fungsional dan nonfungsional yang menunjukan hasil valid pada setiap pengujian yang dilakuan. Pada akhirnya, dapat ditarik kesimpulan bahwa sistem informasi telah dapat dibangun dengan baik.

Kata kunci: Sistem informasi, pengembangan sistem berorientasi objek, penggajian pegawai, dokumentasi pekerjaan.

ABSTRACT

Raga Saputra Heri Istanto, Pengembangan Sistem Informasi Penggajian Pegawai dan Dokumentasi Pekerjaan Berbasis Web (Studi Kasus: PT. Yuga Elektro Tuban)

Pembimbing: Ismiarta Aknuranda dan Mochamad Chandra Saputra

The lack of technology support in the yuga elektro tuban company causing inefficiency in the process of making financial reports and job documentation by administration functionaries. In effort to improve work efficiency, technology support is required that provides the functions to support both process through an information system so that the workload of administration functionaries can be done better.

In order to fulfil the need of technology support through an information system, this research was conducted to determine wheter an protoype of information system capable to demonstrate the functions to support both process can be built properly. To answer those question, requirement analysis, design, and implementation of information system was done using object oriented approach. Then, the result of implementation tested to determine whether system can work as the identified requirement analysis.

The requirement analysis produces the information system requirement identification and visualization of the system capability in the form of use case diagram. The result of design is the object interaction model documentation in the form of sequence diagram, object model in the form of class diagram, data base design, algorithm design, and user interface design. The result is an prototype of information system that capable of demonstrating functions that have been identified in the specification result of information system requirement. Information system then was tested based on functional and nonfunctional requirement that presenting valid results on every test performed. In the end, we can draw a conclusion that information system was built properly.

Key words: Information system, object oriented system development, employee payroll, job documentation.

KATA PENGANTAR

Segala puji bagi Allah SWT atas limpahan rahmat dan karunia yang diberikan sehingga penulis dapat menyelesaikan skripsi berjudul "Pengembangan Sistem Informasi Penggajian Pegawai dan Dokumentasi Pekerjaan Berbasis Web" pada PT. Yuga Elektro Tuban.

Penelitian ini mengajarkan banyak ilmu yang belum saya dapatkan selama kelas perkuliahan. Selain itu, penelitian ini juga mengajarkan saya bagaimana dapat berkomunikasi dengan banyak orang, menyamakan persepsi yang saya miliki dengan pemangku kepentingan dan dosen pembimbing. Proses yang saya alami selama penelitian juga mengajarkan saya untuk menjadi pribadi yang lebih sabar, menurunkan ego, dan senantiasa berbagi hal-hal yang bermanfaat berkaitan dengan kebutuhan pembelajaran. Penelitian ini tentunya tidak bisa terselesaikan tanpa dukungan dan bantuan dari orang-orang terdekat, sehingga saya ingin menyampaikan rasa terimakasih kepada:

1. Ibu saya, Supik yang senantiasa memberikan dukungan doa, moril, materil, mengajarkan ketegasan, dan kedisiplinan dalam menyelesaikan penelitian. Serta senantiasa memberikan contoh yang baik untuk menjadi pribadi yang sabar dan pekerja keras. Ayah saya, Eko Heri Istanto yang telah memberikan kesempatan dan kemudahan untuk dapat melakukan penelitian di lokasi studi kasus dan saya yakini juga memberikan dukungan doa dan menjadi pribadi yang bersahabat dengan saya.
2. Bapak Ismiarta Aknuranda, S.T, M.Sc, Ph.D, selaku pembimbing satu yang mengajarakan ketegasan dan kedisiplinan dalam menyelesaikan kewajiban serta telah menjadi pembimbing yang sabar, solutif dan memberikan contoh sebagai pribadi yang profesional dalam memberikan bimbingan kepada mahasiswa.
3. Bapak Mochamad Chandra Saputra, S.Kom, M.Eng, selaku pembimbing dua yang telah menjadi pembimbing yang sabar, solutif, dan memberikan contoh sebagai pribadi yang profesional dalam memberikan bimbingan kepada mahasiswa.
4. Mirna Suryani, dan segenap pegawai PT. Yuga Elektro Tuban yang memberikan kemudahan pada proses penelitian di lokasi studi kasus.
5. Teman-teman Jurusan Sistem Informasi, Fakultas Ilmu Komputer, yang tidak dapat saya sebutkan namanya satu per satu, yang telah memberikan saya dukungan moril, menjadi teman dan sahabat yang baik dan menyenangkan selama masa perkuliahan.

Semoga Allah SWT melimpahkan rahmat berlipat ganda kepada seluruh pihak yang membantu dan berkontribusi pada penelitian ini. Tidak lupa saya sampaikan bahwa penelitian yang telah dilakukan masih banyak memiliki kekurangan, diskusi

mengenai penelitian lanjutan, saran, dan kritik yang membangun merupakan beberapa hal yang saya harapkan dapat disampaikan kepada saya.

Malang, 7 Juni 2018

Penulis
heriistantoo@gmail.com

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB 1 PENDAHULUAN

1.1 Latar belakang

Yuga Elektro Tuban merupakan perusahaan berskala nasional bergerak di bidang penjualan barang dan jasa konstruksi kelistrikan. Sejak awal berdirinya perusahaan telah banyak mengerjakan proyek baik dari instansi pemerintah maupun swasta. Semakin hari jumlah pekerjaan yang dikerjakan semakin bertambah begitu pula nilai dari pekerjaan tersebut. Pada tahun 2016 perusahaan mengerjakan 7 pekerjaan, pada tahun 2017 bertambah menjadi 9 pekerjaan, dan diperkirakan pada tahun 2018 ini jumlah pekerjaan yang dikerjakan akan bertambah mengingat pada pertengahan tahun jumlah pekerjaan yang dikerjakan sudah mencapai 9 pekerjaan. Bertambahnya jumlah pekerjaan juga diikuti dengan bertambahnya jumlah pegawai yang saat ini sudah mencapai sekitar 130 orang yang terdiri atas pegawai tetap dan pegawai tidak tetap.

Bertambahnya jumlah pekerjaan dan pegawai mengharuskan perusahaan menambah jumlah kantor cabang untuk lebih mudah melakukan manajemen pekerjaannya. Saat ini perusahaan memiliki tiga kantor cabang yang terletak di lokasi berbeda, setiap cabang memiliki petugas admin masing-masing. Setiap petugas admin bertugas untuk mengurus keperluan pekerjaan seperti penggajian pegawai dan pembuatan dokumentasi pekerjaan. Selama ini petugas admin masih melakukan pekerjaanya secara manual sehingga data dari perusahaan tersebar di setiap cabang. Dari analisis hasil wawancara diketahui masalah pada perusahaan muncul ketika tiba saat pembuatan dokumentasi pekerjaan dan laporan keuangan, petugas admin memerlukan waktu yang lama untuk menyelesaikannya. Data yang tersebar pada perusahaan mengakibatkan proses pembuatan laporan keuangan untuk penggajian pegawai serta pembuatan dokumentasi pekerjaan menjadi tidak efisien.

Melihat permasalahan tersebut, sebuah alat bantu yang memiliki fungsi-fungsi untuk pembuatan laporan keuangan dan pembuatan dokumentasi pekerjaan sangat diperlukan guna membantu pekerjaan petugas admin di setiap cabang menjadi lebih efisien. Oleh sebab itu, maksud dari penelitian adalah mengembangkan sebuah alat bantu berupa sistem informasi yang dapat mendemonstrasikan fungsi-fungsi tersebut. Dalam kasus ini alat bantu harus bisa digunakan tanpa harus bergantung pada komputer lokal, mampu mengumpulkan data dalam satu tempat, bisa diakses dari banyak tempat, dan mendukung jalannya proses bisnis perusahaan. Oleh karena itu sistem informasi berbasis web yang didukung dengan data terintegrasi dirasa sangat cocok untuk menyelesaikan permasalahan.

Pengembangan sistem berbasis web untuk mengatasi masalah efisiensi pekerjaan juga pernah dilakukan oleh Herlambang (2017). Penelitian ini bertujuan untuk membangun sistem informasi berbasis web guna menunjukan kemampuan dari model manajemen Andal dengan menerapkan serangkaian tahap analisis

persyaratan, perancangan, dan implementasi berorientasi objek. Model manajemen Andal merupakan model manajemen yang integratif, strategis, efektif, efisien, berkualitas, terjangkau dan merata, dalam pengelolaan pelayanan kesehatan primer pada suatu wilayah. Hasil pengembangan sistem informasi selanjutnya diuji dan menunjukan hasil bahwa sistem informasi yang dibangun dapat berjalan dengan baik.

Menurut Sommerville (2011) dalam mengembangkan sistem bisa digunakan pendekatan *object oriented*. Sistem yang dikembangkan menggunakan pendekatan *object oriented* lebih mudah menghadapi perubahan daripada sistem yang dikembangkan dengan pendekatan fungsional. Ini dikarenakan pada pendekatan *object oriented* mengubah atau menambahkan layanan di suatu objek seharusnya tidak akan berdampak pada objek sistem lainya. Terlebih lagi, karena objek yang berisi data dan operasi merupakan visualisasi dari hal yang berdasar pada dunia nyata, akan memunculkan pemetaan desain yang jelas sehingga meningkatkan pemahaman desain sistem.

Sommervile (2011) juga menyebutkan dalam mengembangkan sistem menggunakan pendekatan *object oriented* terdapat proses pemodelan sistem. Proses pemodelan sistem merupakan suatu proses untuk mengembangkan model abstraks dari suatu sistem dengan setiap model menyajikan pandangan atau prespektif berbeda dari sistem. Dalam proses pemodelan sistem beberapa notasi grafis dapat digunakan sebagai alat bantu merepresentasikan sistem. Pada pendekatan *object oriented* penggunaan notasi grafis dapat didasarkan pada notasi dalam *unified modeling language* (UML). Model UML digunakan pada proses analisis persyaratan untuk membantu memodelkan perysaratan sistem, pada proses perancangan untuk menggambarkan sistem sebagai acuan untuk proses implementasi, dan secara keseluruhan untuk mendokumentasikan struktur dan operasi sistem.

Untuk implementasi sistem yang menerapkan pendekatan *object oriented* digunakan pula bahasa pemrogaman dan kerangka kerja aplikasi yang menerapkan pendekatan *object oriented.* Laravel merupakan salah satu kerangka kerja aplikasi berbasis PHP yang menerapkan pendekatan *object oriented.* Pada saat proses implementasi, objek yang teridentifikasi pada desain diimplementasikan sebagai komponen sistem. Komponen tersebut bertanggung jawab untuk menyediakan fungsi yang merupakan kemampuan dari sistem untuk memenuhi persyaratan pengguna. Untuk memastikan bahwa sistem yang telah dikembangkan memenuhi persyaratan dan untuk mengetahui kekurangan, maka kegiatan pengujian dapat dilakukan.

Berdasarkan penjelasan tersebut, penelitian ini dilakukan guna mengembangkan sebuah prototipe sistem informasi yang dapat mendemonstrasikan fungsi-fungsi untuk mendukung proses penggajian pegawai dan pembuatan dokumentasi pekerjaan dengan melalui serangkaian tahap analisis persyaratan, perancangan, dan implementasi menggunakan pendekatan *object oriented*. Hasil penelitian yang berupa sebuah sistem informasi selanjutnya diuji

sehingga dapat menunjukan hasil bahwa sistem informasi yang telah dikembangkan sesuai dengan persyaratan yang diidentifikasi.

1.2 Rumusan masalah

Berdasarkan paparan masalah yang telah dijelaskan pada latar belakang, secara umum rumusan masalah pada penelitian ini dapat disebutkan sebagai berikut:

Dapatkah dibangun sebuah prototipe sistem informasi yang dapat mendemonstrasikan fungsi-fungsi untuk mendukung proses penggajian pegawai dan dokumentasi pekerjaan?

Untuk menjawab rumusan masalah secara umum, peneliti harus menjawab beberapa rumusan masalah khusus yang telah dirumuskan dan dapat disebutkan sebagai berikut:

1. Bagaimanakah spesifikasi persyaratan sistem berdasarkan analisis data dan hasil wawancara ?
2. Bagaimanakah hasil perancangan sistem berdasarkan spesifikasi persyaratan sistem ?
3. Bagaimanakah hasil implementasi sistem informasi berdasarkan hasil perancangan sistem ?
4. Bagaimanakah hasil pengujian sistem berdasarkan hasil implementasi sistem?

1.3 Tujuan

Di bawah ini disebutkan tujuan secara umum dilakukan penelitian:

Membangun sebuah prototipe sistem informasi yang dapat mendemonstrasikan fungsi-fungsi untuk mendukung proses penggajian pegawai dan dokumentasi pekerjaan.

Tujuan khusus yang harus dipenuhi sebelum mendapatkan tujuan penelitian secara umum adalah:

1. Mengidentifikasi spesifikasi persyaratan sistem informasi berdasarkan analisis data dan hasil wawancara.
2. Merancang sistem informasi sesuai spesifikasi persyaratan.
3. Mengimplementasikan sistem informasi berdasarkan hasil perancangan sistem.
4. Menguji sistem informasi yang telah diimplementasikan.

1.4 Manfaat

Penelitian ini di harapkan mampu mempercepat kinerja dari petugas admin Yuga Elektro Tuban dalam melakukan pembuatan dokumentasi pekerjaan dan laporan keuangaan.

1.5 Batasan masalah

Penelitian ini memilki beberapa batasan masalah yang dapat disebutkan sebagai berikut:

1. Data penelitian merupakan data yang terletak pada cabang perusahaan kecamatan Jatirogo kabupaten Tuban.
2. Sistem informasi difokuskan untuk membantu pembuatan dokumentasi pekerjaan dan laporan keuangan.
3. Sistem informasi berbasis web dibangun dengan bahasa pemrogaman PHP menggunakan kerangka kerja Laravel 4.2.

1.6 Sistematika pembahasan

BAB 1 PENDAHULUAN

Bab ini berisi sistematikan pembahasan penelitian, batasan masalah, manfaat penelitian, rumusan masalah, tujuan penelitian, dan latar belakang masalah.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini berisi pembahasan beberapa teori yang terapkan pada penelitian. Dalam bab ini juga disebutkan beberapa penelitian yang telah dilakukan sebelumnya sebagai referensi dalam melakukan penelitian.

BAB 3 METODOLOGI

Pada bab ini dijelaskan alur penelitian yang dilakukan oleh peneliti untuk menyelesaikan penelitian.

BAB 4 PEMODELAN PROSES BISNIS & ANALISIS PERSYARATAN SISTEM

Bab ini menjelaskan hasil analisis persyaratan sistem yang terdiri dari analisis masalah, identifikasi tipe pemangku kepentingan, pemodelan proses bisnis, identifikasi pengguna, identifikasi kebutuhan pemangku kepentingan dan pengguna, identifikasi fitur, pemodelan aktivitas, dan pemodelan *use case.*

BAB 5 PERANCANGAN

Bab ini menjelaskan hasil perancangan sistem ke dalam bentuk *class diagram* secara umum, *class diagram* kerangka kerja pemrogaman, perancangan basis data ke dalam bentuk PDM, perancangan antarmuka, dan perancangan algoritme.

BAB 6 IMPLEMENTASI

Bab ini menjelaskan hasil implementasi penelitian berupa spesifikasi lingkungan *deployment,* spesifikasi lingkungan

implementasi, beberapa implementasi antarmuka pengguna, dan beberapa implementasi algoritme.

BAB 7 PENGUJIAN

Bab ini menjelaskan hasil pengujian sistem dengan menggunakan metode *basis path,* validasi, kompatibilitas.

BAB 8 PENUTUP

Kesimpulan serta saran dari penelitian dijelaskan pada bab ini.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Penelitian Terdahulu

Penelitian ini berdasar pada beberapa referensi penelitian terdahulu tentang penggunaan pendekatan berorientasi objek dalam melakukan pengembangan sistem. Penelitian terdahulu juga menjadi landasan peneliti untuk memanfaatkan beberapa diagram *Unified Modeling Language* (UML) dalam aktivitas menganalisis persyaratan dan perancangan sistem. Hasil analisis persyaratan dan perancangan kemudian menjadi dasar dalam melakukan implementasi sistem informasi yang menerapkan pendekatan *object oriented* mengikuti perancangaan yang telah dilakukan sebelumnya.

Herlambang (2017) pada penelitianya menunjukan bahwa sistem informasi berbasis web untuk mendukung model manajemen Andal yang dibangun dengan menerapkan serangkaian tahap analisis persyaratan, perancangan, dan implementasi berorientasi objek dapat dibangun dengan baik dan mampu mendukung jalanya model manajemen yang integratif, strategis, efektif, efisien, berkualitas, terjangkau dan merata dalam pengelolaan pelayanan kesehatan primer pada suatu wilayah.

Pengembangan sistem berbasis web untuk mengatasi masalah efisiensi pekerjaan juga pernah dilakukan oleh Herlambang (2017). Penelitian ini bertujuan untuk membangun sistem informasi berbasis web guna menunjukan kemampuan dari model manajemen Andal dengan menerapkan serangkaian tahap analisis persyaratan, perancangan, dan implementasi berorientasi objek. Model manajemen Andal merupakan model manajemen yang integratif, strategis, efektif, efisien, berkualitas, terjangkau dan merata, dalam pengelolaan pelayanan kesehatan primer pada suatu wilayah. Hasil pengembangan sistem informasi selanjutnya diuji dan menunjukan hasil bahwa sistem informasi yang dibangun dapat berjalan dengan baik.

Sommerville (2011) menjelaskan bahwa dalam pengembangan sistem bisa digunakan pendekatan *object oriented* yang memiliki keunggulan untuk lebih mudah menghadapi perubahan daripada sistem yang dikembangkan menggunakan pendekatan fungsional. Pengembangan sistem menggunakan pendekatan *object oriented* juga akan memunculkan pemetaan desain yang jelas sehingga mempermudah pemahaman desain sistem. Sommerville (2011) juga menyebutkan dalam melakukan pengembangan sistem dengan pendekatan *object oriented* terdapat proses pemodelan sistem. Pemodelan sistem berarti mewakili sistem menggunakan beberapa notasi grafis. Pada pengembangan sistem menggunakan pendekatan *object oriented* penggunaan notasi grafis dapat didasarkan pada notasi dalam *unified modeling language* (UML). Model UML digunakan pada proses analisis persyaratan untuk membantu memodelkan persyaratan sistem, pada proses perancangan untuk menggambarkan sistem sebagai acuan untuk proses implementasi sistem, dan secara keseluruhan juga untuk mendokumentasikan

struktur dan operasi sistem. Penjelasan tersebut menjadi dasar peneliti menggunakan UML sebagai bahasa pemodelan pada penelitian ini.

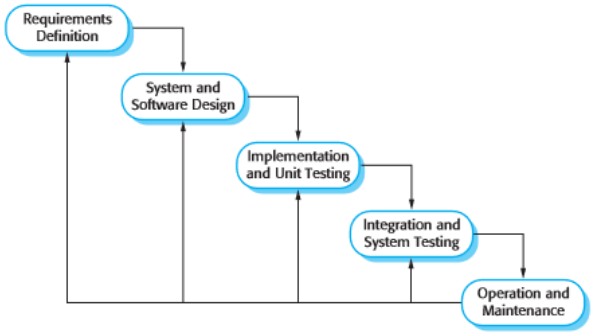
Hasil penelitian Odeh et al (2003) menunjukan bahwa penggunaan teknik *use case* mampu memperpendek jarak antar setiap fase pada proses analisis persyaratan, sehingga proses analisis persyaratan menjadi lebih cepat dilakukan. Berdasarkan paparan penjelasan dan manfaat dari penelitian tersebut, teknik *use case* digunakan oleh peneliti dalam kegiatan analisis persyaratan sistem informasi.

Adam Przybyłek (2011) pada penelitiannya menjelaskan bahwa terdapat teknik yang bisa digunakan untuk menyelaraskan antara *busines process model* dengan diagram *use case* sehingga *use case* yang dibuat mampu sepenuhnya mendukung proses bisnis dari perusahaan. Penelitian tersebut menunjukan teknik yang mendukung ketelurusan perpindahan dari *business process model* menjadi diagram *use case*, sehingga *use case* yang dihasilkan akan sesuai dengan proses bisnis usulan yang berjalan pada perusahaan. Hasil penelitian tersebut menjadi dasar untuk menerapkan teknik yang mampu mendukung ketelurusan antara model proses bisnis dengan diagram *use case*. Teknik ini didokumentasikan dengan menambahkan tabulasi yang menunjukan hubungan aktivitas pada proses bisnis usulan dengan *use case* yang terindikasi pada penelitian ini.

Marina Krol dan David L. Reich (1999) dalam penelitiannya menjelaskan keuntungan analisis dan perancangan perangkat lunak menggunakan pendekatan *object oriented* bahkan pada sistem komputer yang kompleks. Penelitian ini berfokus pada penggunaan objek (termasuk identitas, hubungan, atribut, dan operasi) pada perancangan sistem. Setiap kelas mendefinisikan nilai atribut dan operasi yang dibawa oleh setiap objek tersebut. Penelitian ini mengungkapkan analisis dan perancangan menggunakan *object oriented* mampu menambah konsistensi dan usabilitas. Hasil penelitian ini memperkuat peneliti menggunakan pendekatan *object oriented* pada penelitian secara umum atau secara khusus pada analisis persyaratan dan perancangan sistem informasi.

2.2 Waterfall *Model*

Merupakan prinsip pengembangan sistem berbasis rencana di mana semua proses aktivitas pengembangan perangkat lunak harus sudah terencana dan terjadwal sebelum dikerjakan (Sommerville, 2011). Tahapan yang terjadi pada pengembangan perangkat lunak menggunakan model *waterfall* bisa ditunjukan pada Gambar 2.1.



Gambar 2.1 *Waterfall model*

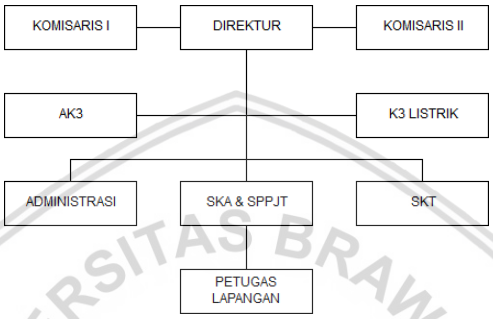
Sumber: Sommerville (2011)

1. *Requirements definition.* Pada tahap ini dilakukan diskusi dengan para pengguna untuk menentukan batasan, tujuan, dan layanan sistem. Temuan yang diperoleh dari kegiatan diskusi akan dijadikan bahan untuk membuat dokumen persyaratan.
2. *System and software design.* Proses perancangan sistem dengan cara membangun arsitektur sistem secara keseluruhan berdasarkan dokumen persyaratan yang telah dibuat.
3. *Implementation and unit testing.* Proses untuk merealisasikan peracangan sistem ke dalam serangkaian program atau unit program. Pada tahap ini juga dilakukan pengujian untuk mengetahui bahwa setiap unit program telah sesuai dengan dokumen persyaratan sistem.
4. *Integration and system testing.* Tahap untuk mengintegrasikan setiap unit program sehingga menjadi satu kesatuan yang utuh. Hasil integrasi unit program juga diuji untuk memastikan sistem yang telah di kembangkan memenuhi persyaratan. Selanjutnya, sistem diserahkan kepada pelanggan.
5. *Operation and maintenance.* Pada tahap ini sistem telah terpasang dan digunakan oleh pelanggan. Perawatan meliputi memperbaikan kesalahan sistem yang tidak ditemukan pada saat masa awal pengembangan perangkat lunak, meningkatkan kinerja sistem, dan menambahkan fungsionalitas sistem ketika persyaratan yang baru ditemukan.

Dalam bukunya Sommerville juga menambahkan, tahap berikutnya seharusnya tidak dimulai sampai tahap sebelumnya selesai. Namun pada prakteknya, fase yang terjadi pada proses pengembangan perangkat lunak saling tumpang tindih dan memberi informasi satu sama lain, menjadikan model ini bukanlah model linier sederhana namun melibatkan umpan balik dari satu fase ke fase lainya. Dokumen yang diproduksi pada setiap fase kemudian harus dimodifikasi untuk mencerminkan perubahan yang terjadi.

2.3 Yuga Elektro Tuban

Yuga Elektro Tuban merupakan organisasi penyedia jasa ketenagalistrikan yang memiliki wilayah kerja di seluruh Indonesia. Struktur organisasi Yuga Elektro Tuban dapat ditunjukan pada Gambar 2.2. Berdasarkan Gambar 2.2, Yuga Elektro Tuban dipimpin oleh seorang Direktur yang diawasi oleh 2 orang Komisaris. Direktur membawahi 5 unit dan 1 subunit, yaitu AK3, K3 Listrik, Administrasi, SKT (unit sertivikat keterampilan kerja), dan SKA & SPPJT (unit penanggung jawab teknik) yang juga membawahi unit Petugas Lapangan.



Gambar 2.2 Struktur organisasi Yuga Elektro Tuban

2.4 Proses Bisnis

Proses bisnis merupakan serangkaian aktivitas yang membutuhkan sebuah *input* dan menghasilkan *output* serta memiliki nilai bagi *customer* (Monk & Wagner, 2013). Pada proses bisnis terdapat dua jenis *customer* yaitu *customer internal* dan *customer eksternal*. *Customer internal* merupakan seseorang atau sekelompok orang yang berada pada bermacam department dalam perusahaan. *Customer eksternal* merupakan seseorang atau sekelompok orang yang membeli produk. Pemodelan proses bisnis dapat digunakan untuk meningkatkan pemahaman terhadap sebuah perusahaan atau organisasi. Terdapat beberapa pendekatan yang dapat digunakan untuk meningkatkan pemahaman, diantaranya adalah dengan cara melakukan analisis dan pemodelan terhadap proses bisnis *as-is* dan *to-be.*

2.4.1 Analisis dan Pemodelan Proses Bisnis *As-Is*

Merupakan suatu kegiatan untuk memodelkan proses bisnis aktual pada perusahaan atau organisasi. Kegiatan ini bertujuan untuk mengetahui kekurangan proses bisnis sehingga dapat memunculkan kemungkinan peningkatan model proses bisnis yang diterapkan pada perusahaan atau organisasi (Becker, et al., 2013).

Terdapat beberapa metode yang dapat digunakan untuk mengidentifikasi proses bisnis *as-is* seperti melakukan observasi terhadap pegawai yang sedang melakukan tugasnya atau melakukan wawancara terhadap individu atau kelompok yang berpartisipasi dalam berlangsungnya proses bisnis. Setiap aktivitas proses bisnis yang diketahui berdasarkan metode tersebut akan dimodelkan. Aktivitas

proses bisnis yang kompleks dapat dipecah sebagai sub-proses untuk memudahkan pemahaman terhadap model (Przbytek, 2011).

2.4.2 Analisis dan Pemodelan Proses Bisnis *To-be*

Merupakan aktivitas mengidentifikasi dan memodelkan proses bisnis yang telah megalami peningkatan berdasarkan analisis proses bisnis *as-is*. Untuk dapat memperoleh kemungkinan perbaikan atau peningkatan proses bisnis, maka pemangku kepentingan perlu dilibatkan untuk memberikan masukan dan pengetahuan terhadap proses bisnis. Peningkatan proses bisnis bisa didapatkan dengan mengotomasi aktivitas dalam satu atau beberapa proses bisnis (Przbytek, 2011). Pada perusahaan atau organisasi penerapan proses bisnis *to-be* dapat dilakukan dalam waktu menengah atau panjang dan dilakukan secara bertahap.

2.5 *Business Process Modelling Notation* (BPMN)

Merupakan standar notasi yang digunakan untuk melakukan pemodelan proses bisnis sehingga proses pertukaran informasi antar pengguna bisnis dapat dimodelkan dengan baik (Object Management Group, 2011). BPMN dapat digunakan untuk analisis persiapan proses bisnis baru. Pada BPMN terdapat beberapa notasi yang dapat digunakan untuk melakukan pemodelan proses bisnis, diantaranya adalah *data, event, activity, Flow, gateway, swimlanes,* dan *artifacts.*

2.5.1 Data

Merupakan notasi yang memberikan visualisasi terhadap dokumen, objek, atau data yang diperbaruai dan digunakan pada sebuah alur proses. Kategori notasi data dapat dibedakan kembali menjadi beberapa notasi seperti *data object* dan *data stores*. Penjelasan kedua notasi data dapat ditunjukan pada Tabel 2.1.

Tabel 2.1 Notasi *data*

Notasi	Nama	Deskripsi
	*Data Object*	Notasi yang merupakan visualisasi terhadap dokumen, objek, atau data yang dimanfaatkan oleh proses atau aktivitas tertentu pada proses bisnis.
	*Data Store*	Merupakan notasi yang menjelaskan sebuah lokasi penyimpanan dan pembaruan data pada suatu proses bisnis.

Sumber: Object Management Group (2011)

2.5.2 Flow

Pada pemodelan proses bisnis hubungan antar notasi dapat dihubungkan menggunakan notasi *flow*. Penelitian ini memanfaatkan beberapa notasi *flow* yang dapat dijelaskan pada Tabel 2.2.

Tabel 2.2 Notasi *flow*

Notasi	Nama	Deskripsi
	Sequence Flow	Merupakan visualisasi urutan aktivitas yang dijalankan pada proses bisnis.
	Message Flow	Notasi untuk menunjukan aliran pesan masuk dan keluar pada partisipan yang berbeda.
	Associations	Merupakan notasi yang menghubungkan *artifact* dengan notasi lainya.
	Data Associations	Visualisasi dari notasi yang menghubungkan notasi data dengan notasi lain.

Sumber: Object Management Group (2011)

2.5.3 Events

Merupakan notasi yang berpengaruh terhadap berjalanya alur aktivitas proses bisnis. Notasi ini dapat dikategorikan ke dalam tiga kategori yaitu *intermediate event, end event,* dan *start event*. Pada penelitian ini ketiga kategori tersebut digunakan. Penjelasan dari ketiga kategori tersebut dapat dilihat pada Tabel 2.3.

Tabel 2.3 Notasi *event*

Notasi	Nama	Deskripsi
	Start Event	Merupakan nota yang menggambarkan awal dimulainya suatu proses.
	Intermediate Event	Notasi yang berada antara *start event* dan *end event* ini merupakan notasi yang memberikan pengaruh terhadap alur proses bisnis. Notasi ini juga memiliki beberapa varian yang ditandai dengan penggunaan simbol berbeda ditengah notasi ini.
	End	Visualisasi berhentinya proses.

Sumber: Object Management Group (2011)

2.5.4 Activity

Merupakan notasi yang menggambarkan aktivitas pada suatu proses. Notasi ini memiliki dua jenis yaitu *task* dan *sub process*. Pada penelitian ini hanya digunakan jenis *task*. Penjelasan singkat tentang notasi *activity* yang digunakan pada penelitian ini dapat ditunjukan pada Tabel 2.4.

Tabel 2.4 Notasi *activity*

Notasi	Nama	Deskripsi
	Task	Merupakan visualisasi dari aktivitas tunggal yang tidak bisa dipecah atau didetailkan lagi.

Sumber: Object Management Group (2011)

2.5.5 Gateway

Merupakan notasi untuk menentukan alur berjalanya suatu proses bisnis. Notasi ini memeiliki beberapa varian yang dapat ditunjukan pada Tabel 2.5.

Tabel 2.5 Notasi *gateways*

Notasi	Nama	Deskripsi
	Exclusive	Notasi yang menggambarkan alur alternatif. Hanya akan terpilih satu alur alternatif untuk dijalankan sesuai dengan kondisi yang terpenuhi.
	Parallel	Notasi yang merupakan visualisasi alur paraler pada suatu proses.
	Inclusive	Digunakan untuk membuat alur alternatif dan paraler sekaligus pada suatu proses. Setiap alur dengan kondisi terpenuhi akan dikerjakan.

Sumber: Object Management Group (2011)

2.5.6 Swimmlanes

Dalam pemodelan proses bisnis pengelompokan aktivitas dapat dilakukan berdasarkan unit bisnis menggunakan notasi *swimmlanes*. Penjelasan notasi *swimmlanes* yang digunakan pada penelitian ini dapat ditunjukan pada Tabel 2.6.

Tabel 2.6 Notasi *swimmlanes*

Notasi	Nama	Deskripsi
	Pool	*Pool* merupakan notasi yang dapat memberikan visualisasi aktivitas yang dimiliki oleh seluruh partisipan.
	Lane	Notasi yang memvisualisasikan subpartisi dari suaut *pool*. Dalam satu lane hanya terdapat sebuah partisipan.

Sumber: Object Management Group (2011)

2.5.7 Artifacts

Artifacts dapat digunakan sebagai visualisasi informasi tambahan pada suatu notasi di dalam proses bisnis. Penggunaan *artifacts* hanya bertujuan untuk memperjelas notasi yang digunakan dalam model proses bisnis, sehingga tidak

memiliki pengaruh apapun terhadap alur proses bisnis. Terdapat beberapa notasi *artifacts* yang dapat digunakan seperti yang bisa dilihat pada Tabel 2.7.

Tabel 2.7 Notasi *artifacts*

Notasi	Nama	Deskripsi
	Group	Notasi *group* dapat digunakan untuk mengelompokan beberapa notasi ke dalam satu kategori. Nama kelompok dapat di tuliskan pada bagian bawah notasi. Tidak ada alur yang terpengaruh dari penggunaan notasi ini.
	Text Annotation	Merupakan notasi yang memberikan visualisasi informasi tambahan pada notasi lain sehingga meningkatkan pemahaman pembaca terhadap model.

Sumber: Object Management Group (2011)

2.6 Pemodelan *Use Case*

Pemodelan *use case* mampu memberikan manfaat besar di dalam aktivitas analisis persyaratan sistem pada proses pengembangan perangkat lunak (Bittner & Spence, 2002). Penggunaan *use case* mampu mempermudah komunikasi antara pemangku kepentingan dengan tim pengembang perangkat lunak. Penggunaan *use case* juga mampu membuat proses penggambaran pengguna menggunakan sistem dan apa yang dilakukan sistem untuk pengguna menjadi lebih tepat dan mudah.

Pemanfaatan *use case* juga mampu memberikan manfaat lain diantaranya adalah mengetahui informasi pemangku kepentingan, mengetahui kebutuhan pemangku kepentingan berdasarkan masalah yang dihadapi, mengetahui kemungkinan solusi yang dapat diusulkan kepada pemangku kepentingan untuk memenuhi kebutuhanya, serta mengetahui pengguna sistem.

2.6.1 Identifikasi Tipe Pemangku Kepentingan

Merupakan kegiatan untuk mengetahui siapa saja pemangku kepentingan yang terdapat pada suatu pengembangan sistem. Menurut (Bittner & Spence, 2002) pemangku kepentingan merupakan seseorang atau sekelompok orang yang secara materi terkena dampak dari pengembangan sistem. Pada prakteknya setiap pemangku kepentingan yang teranalisis akan dikategorikan berdasarkan karakteristik dan hubunganya dengan sistem yang dikembangkan. Terdapat beberapa kategori pemangku kepentingan yang digunakan dalam penelitian ini, di antaranya adalah:

1. *Users* (Pengguna), merupakan seseorang yang berinteraksi secara langsung dengan sistem. Tipe ini akan berberan sebagai *aktor* pada pemodelan *use case*.

2. *Developers* (Pengembang), tipe yang merupakan pelaku utama kegiatan produksi perangkat lunak.
3. *Authorities* (Pihak yang berwenang), merupakan pemberi dukungan di dalam proses pengembangan sistem. Dukungan yang diberikan dapat berupa standar organisasi, aturan perundangan, teknologi, dan regulas internal atau eksternal perusahaan.
4. *Customers* (Pelanggan), tipe yang mendapatkan manfaat dari hasil pengembangan sistem.

Pada dasarnya tipe pemangku kepentingan tidak terikat pada tipe-tipe yang telah dijelaskan di atas. Tipe pemangku kepentingan dapat dikembangkan dengan bebas mengikuti macam calon pengguna sistem, jenis perusahaan atau organisasi, dan faktor-faktor pendukung lainya.

2.6.2 Analisis Masalah

Merupakan serangkaian kegiatan memahami kebutuhan dan masalah aktual pemangku kepentingan. Dari pemahaman tersebut muncul usulan sebagai solusi dari kebutuhan pemangku kepentingain. Dalam melakukan kegiatan analisis masalah perlu adanya suatu dokumentasi. Menurut (Bittner & Spence, 2002) dalam menyusun dokumentasi analisis masalah dapat dituliskan *problem statement* (pernyataan masaalah) mengikuti kerangka penulisan dokumentasi yang dapat ditunjukan pada Tabel 2.8. Dalam mengumpulkan pernyataan masalah dapat digunakan teknik wawancara terhadap beberapa pemangku kepentingan.

Tabel 2.8 Kerangka dokumentasi pernyataan masalah

The problem of (Masalah)	Penulisan masalah yang dimiliki oleh pemangku kepentingan.
Affects (Mempengaruhi)	Penulisan pada pemangku kepentingan mana masalah tersebut berpengaruh.
The impact of which is (Dampak)	Penulisan dampak apa yang dihasilkan oleh masalah terhadap kegiatan bisnis dan pemangku kepentingan.
A succesful solution would (Solusi)	Berisi solusi untuk menyelesaikan masalah yang dihadapi oleh pemangku kepentingan.

Sumber: Bittner & Spence (2002)

2.6.3 Identifikasi Kebutuhan Pemangku Kepentingan dan Pengguna

Merupakan aktivitas untuk mendapatkan informasi kebutuhan pemangku kepentingan dan pengguna sehingga mampu memahami lebih baik bagaimana suatu masalah dapat mempengaruhi pemangku kepentingan (Bittner & Spence, 2002). Dengan pemahaman masalah yang baik maka akan memperbesar munculnya solusi pada setiap kebutuhan yang dimiliki oleh pemangku kepentingan dan pengguna.

2.6.4 Identifikasi Fitur

Fitur merupakan ringkasan kemampuan atau layanan yang disediakan sistem sebagai cara pemenuhan kebutuhan pemangku kepentingan. Dengan mengidentifikasi fitur maka fungsionalitas dari sistem sudah dapat sedikit dijelaskan tanpa perlu terhambat penjelasan yang mendetail (Bittner & Spence, 2002). Dalam melakukan identifikasi fitur pemberian nama fitur harus semaksimal mungkin dapat dipahami oleh seluruh pemangku kepentingan. Hal ini bermaksud, nama dari sebuah fitur harus memiliki tingkat detail yang umum. Fitur yang teridentifikasi selanjutnya didokumentasikan ke dalam tabulasi mengikuti kerangka yang terdapat pada Tabel 2.9.

Tabel 2.9 Contoh tabel hasil identifikasi fitur

Kode Fitur	Deskripsi
Identitas fitur	Penjelasan singkat fitur

Sumber: Diadaptasi dari Bittner & Spence (2002)

2.6.5 Identifikasi Persyaratan Fungsional dan Persyaratan Nonfungsional

Merupakan cara yang dilakukan untuk melakukan pengelempokan persyaratan sistem. Pada umumnya persyaratan sistem dapat dibagi menjadi dua jenis yaitu, fungsional dan nonfungsional. Untuk mendokumentasikan hasil identifikasi, persyaratan dapat dipresentasikan ke dalam suatu kalimat deklaratif (Bittner & Spence, 2002).

Persyaratan nonfungsional adalah persyaratan yang mendefinisikan batasan atau kualitas yang harus dipenuhi oleh sistem. Persyaratan nonfungsional muncul karena pentingnya penentuan aspek *compatibility, performance, reliability, usability* di dalam suatu sistem.

Persyaratan fungsional lebih kepada tindakan yang mampu dilakukan oleh sistem dan sebaiknya disediakan oleh sistem. Persyaratan fungsional berfokus untuk menjelaskan bagaimana reaksi sistem terhadap *input* dan bagaimana sistem berjalan pada suatu situasi tertentu (Sommerville, 2011).

2.6.6 Pemodelan *Use Case*

Merupakan model yang digunakan sebagai visualisasi penggunaan sistem dan visualisasi interaksi antara pengguna dengan sistem (Bittner & Spence, 2002). Pada pemodelan *use case* terdapat aktor dan *use case* sebagai komponen utama. Dalam menentukan *use case* dapat digunakan hasil analisis fitur yang telah ditentukan.

Use case tidak sama dengan sebuah fungsi dari sistem. *Use case* lebih kepada pemberian nilai oleh sistem kepada aktor. Setiap *use case* memiliki nama dan penjelasan singkat. Di dalam masing-masing *use case* terdapat spesifikasi yang menjelaskan alur interaksi yang dilakukan aktor dengan *use case* untuk memenuhi kebutuhan aktor. Terdapat beberapa cara yang dapat digunakan dalam

mendokumentasikan spesifikasi *use case*. Format dokumentasi spesifikasi *use case* pada penelitian ini dapat dijelaskan pada Tabel 2.10.

Tabel 2.10 Format dokumentasi spesifikasi *use case*

Brief Description	Merupakan penjelasan singkat sebuah *use case.*
Actor	Merupakan nama aktor pelaku interaksi.
Basic Flow	Penjelasan alur normal berjalanya sebuah *use case*.
Alternative Flow	Penjelasan alur opsional yang dilalui oleh *use case* ketika tidak berjalan sebagaimana alur normal.
Subflow	Merupakan penyerdeharnaan alur *use case* yang detail menjadi beberapa alur yang lebih sederhana sehingga spesifikasi lebih mudah dipahami.
Pre-Condition	Merupakan penjelasan dari satu atau lebih kondisi yang harus terpenuhi sebelum alur *use case* dapat berjalan.
Post-Condition	Merupakan penjelasan satu atau lebih kondisi yang dimiliki sistem pada saat *use case* berakhir.

Sumber: Diadaptasi dari Bittner & Spence (2002)

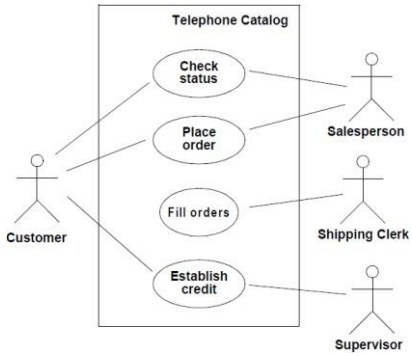
Aktor merupakan objek diluar sistem yang memberikan visualisasi individu yang melakukan interaksi dengan sistem. Pada aktor melekat nama dan penjelasan singkat. Pada pemodelanya, aktor selalu dihubungkan dengan *use case* mengikuti interaksi yang terjadi.

2.7 *Unified Modeling Language* (UML)

Merupakan bahasa yang digunakan untuk mendokumentasikan, memvisualkan, membangun dan menentukan artefak perangkat lunak (Object Management Group, 2005). Banyak varian yang terdapat pada UML seperti *behaviour diagram, implementation diagram, class diagram,* dan *use case diagram*. Setiap jenis diagram menggambarkan pandangnya masing-masing, sehingga kombinasi beberapa jenis diagram bisa digunakan untuk memodelkan sistem dari berbagai sudut pandang. Pada penelitian ini dimanfaatkan beberapa jenis diagram diantaranya *sequence diagram, activity diagram* yang merupakan varian dari *behaviour diagram*, *class diagram,* dan *use case diagram.*

2.7.1 Use Case Diagram

Diagram *use case* menggambarkan hubungan antara satu atau lebih *use case* dengan aktor. Terdapat juga hubungan generalisasi antar aktor, hubungan *extends,* dan hubungan *includes.* Sekumpulan *use case* memiliki batasan sistem yang digambarkan dengan notasi berbentuk persegi panjang. Berikut pada Gambar 2.3 merupakan contoh pemodelan diagram *use case.*

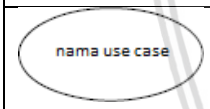
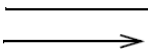
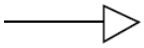


Gambar 2.3 Contoh *use case* diagram

Sumber: Object Management Group (2005)

Pada penelitian ini terdapat beberapa notasi yang digunakan di antaranya adalah notasi aktor, notasi *use case,* dan relasi. Penjelasan dari beberapa notasi tersebut dapat ditunjukan pada Tabel 2.11.

Tabel 2.11 Notasi *use case diagram*

Notasi	Nama	Deskripsi
	*Actor*	Sebuah aktor bisa berinteraksi dengan satu atau lebih *use case.* Menurut pengertianya aktor merupakan peran yang diperankan oleh pengguna atau sistem lain yang berinteraksi dengan sistem.
	*Use case*	Merupakan nilai yang diberikan oleh sistem untuk aktor yang berisi sekumpulan aksi.
	*Association*	Notasi untuk menggambarkan interaksi antara *use case* dengan aktor. Menggunakan notasi ini satu aktor dapat melakukan interaksi terhadap satu atau lebi *use case*.
	*Generalization*	Notasi yang digunakan untuk menggambarkan hubungan aktor spesifik kepada aktor yang lebih umum.

Sumber: Diadaptasi dari Object Management Group (2005)

2.7.2 Class Diagram

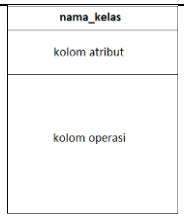
Merupakan pemodelan yang dilakukan untuk menggambarkan struktur, deksripsi dan hubungan antar kelas di dalam suatu perangkat lunak. *Class diagram* bisa digunakan pada saat proses pengembangan perangkat lunak khususnya pada bagian perancangan. Pada tahap tersebut pemodelan kelas dapat digunakan untuk memodelkan objek dari sebuah sistem (Boch, et al, 1998). Berikut beberapa pendekatan yang bisa dilakukan dalam melakukan pemodelan kelas dengan *class diagram* :

1. *Modelling a Logical Database Schema*

Pendekatan ini menggunakan *class* diagram untuk menggambarkan rancangan basis data. Rancangan yang dimodelkan merepresentasikan objek yang akan menyimpan data sistem.

Penjelasan mengenai notasi pada *class diagram* bisa dijelaskan pada Tabel 2.12.

Tabel 2.12 Notasi *class diagram*

Notasi	Nama	Deskripsi
	*Class*	Merupakan notasi yang memberikan visualisasi objek beserta atribut dan operasinya. Memiliki tiga buah kolom yang diperuntukan bagi atribut, operasi, dan nama kelas.
	Generalization	Notasi yang digunakan untuk menghubungkan ojek yang umum (*superclass*) dengan objek yang lebih spesifik (*subclass)*. Relasi antara *subclass* dengan *superclass* akan mewariskan atribut dan operasi yang d imiliki *superclass* kepada *subclass.*
-	*Private (Visibility)*	Notasi yang digunakan untuk merepresentasikan aksesibilitas atribut atau operasi yang dimiliki oleh objek. Aksesibilitas *private* hanya dapat diakses oleh objek yang memiliknya.
+	*Public (Visibility)*	Aksesibilitas *public* merepresentasikan visibilitas yang dapat secara langsung diakses oleh objek manapun.
	Composition	Relasi untuk menunjukan hubungan antara objek yang lebih luas (*whole)* terhadap objek yang menjadi bagian dari objek tersebut (*parts*). Objek *whole* dilekatakan pada sisi ujung belah ketupat dan objek *part* diletakan pada sisi lainya. Pada *composition* objek *part* merupakan bagian dari hanya satu objek *whole.*

Sumber: Diadaptasi dari Object Management Group (2005)

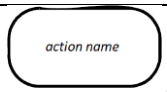
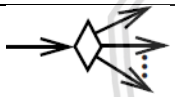
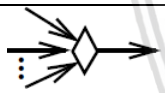
2. *Modelling the Vocabulary of a System*

Memodelkan kelas dari sebuah sistem yang didapatkan dari suatu masalah yang ingin diselesaikan melalui sistem yang akan dikembangkan. Setiap kelas dimodelkan ke dalam notasi kelas-kelas beserta atribut dan kemampuan yang dapat dilakukan oleh kelas tersebut. Kolaborasi antar kelas juga bisa digabungkan untuk merepresentasikan hubungan antar objek yang bekerja sama untuk menunjukan perilaku sistem.

2.7.3 *Activity Diagram*

Merupakan pemodelan yang digunakan untuk menggambarkan urutan langkah dalam proses komputasi. Pemodelan ini lebih menekankan pada pemodelan alur antar aktivitas dan kolaborasi antara aktor dengan sistem. *Activity diagram* mampu mempermudah pemangku kepentingan dalam memahami alur kolaborasi (Bittner & Spence, 2002). Dalam pemodelanya *activity diagram* memiliki beberapa notasi yang bisa digunakan seperti yang dapat dijelaskan pada Tabel 2.13:

Tabel 2.13 Notasi pada *activity diagram*

Notasi	Nama	Deskripsi
	*Initial Node*	Notasi memberikan visualisasi awal dimulainya suatu aktivitas yang terdapat pada model.
	*Final Node*	Merupakan tanda titik berakhirnya alur aktivitas.
	*Actions*	Penggambaran tindakan yang dilakukan oleh sebuah unit bisnis.
	Activity Edges	Penghubung antar notasi di dalam model.
	*Decision Node*	Digunakan sebagai penggambaran alur kondisional dari edges. Notasi ini tidak merepresentasikan alur sinkronisasi atau alur paralel, sehingga alur hanya akan berjalan melalui satu *edge* keluar sesuai kondisi yang terpenuhi.
	*Merge Node*	Digunakan untuk menerima beberapa alur alternatif dengan kondisi terpenuhi.
	*Activity Partition*	Notasi yang digunakan untuk mengelompokan aktivitas.

Sumber: Diadaptasi dari Object Management Group (2005)

2.7.4 *Sequence Diagram*

Merupakan visualisasi urutan pertukaran pesan yang terjadi antar objek untuk mendapatkan hasil tertentu berdasarkan urutan waktu (Leffingwell & Widrig, 2003). Pada diagram ini terdapat beberapa notasi yang merepresentasikan antarmuka, kelas, dan pertukaran pesan yang terjadi diantara notasi tersebut. Beberapa notasi yang digunakan pada penelitian ini dapat dijelaskan pada Tabel 2.14.

Tabel 2.14 Notasi pada *sequence diagram*

Notasi	Nama	Deskripsi
	*Actor*	Merupakan pelaku yang diperankan oleh pengguna atau sistem lain.

sd EventOccurrence	*Frame*	Notasi yang merupakan tempat untuk serangkaian objek yang saling bertukar pesan. Nama dari interaksi dituliskan pada bagian pojok kiri atas di dalam bangun segilima.
:Lifeline	*Lifeline*	Notasi yang merupakan visualisasi dari objek yang terlibat pada pertukaran pesan dalam urutan waktu.
message_name	*Message*	Notasi yang merepresentasikan komunikasiantar objek. Jika komunikasi dengan memanggil operasi berargumen, maka argumen ikut disertakan dalam penamaan notasi.
message_name	*Message reply*	Notasi yang menggambarkan pesan balasan dari penggunaan notasi *message.*
alt	*Combined Fragment*	Merupakan notasi untuk mengelompokan interaksi ke dalam suatu *fragment*. Setiap *fragment* memiliki beberapa *operand* yang masing-masing ditempatkan pada jalur yang dipisahkan oleh garis putus-putus di dalam *fragment*. Setiap *operand* memiliki sebuah *guard* yang menandakan kondisi yang dipenuhi ketika *operand* tersebut dijalankan. Notasi ini memiliki beberapa varian yang ditandai dengan penamaan *interaction operator* di bangun segi lima pada pojok kiri atas notasi. Varian *loop* merupakan varian yang merepresentasikan interaksi yang dilakukan secara berulang-ulang selama kondisi *operand* terpenuhi. Varian ini hanya dapat memiliki satu *operand*.

Sumber: Diadaptasi dari Object Management Group (2005)

2.8 *Physical Data Model* (PDM)

PDM merupakan pemodelan basis data yang konsepnya hampir serupa dengan pemodelan kelas, namun tujuan utamanya adalah untuk menggambarkan hubungan antar tabel beserta kolomnya serta memodelkan skema internal basis data. Pemodelan basis data dapat dilakukan berdasarkan hasil pemodelan kelas dalam *class diagram*. Pemodelan basis data menggunakan PDM dapat dilakukan dengan salah satu atau kombinasi dari tiga strategi (Botch, et al), yaitu:

1. Memisahkan *subclass* dengan *superclass* dan mendefinisikanya menjadi tabel pada PDM. Pendekatan ini memiliki kekurangan yaitu memerlukan mekanisme *join table* untuk proses pengambilan data melalui basis data.

2. Mendefinisikan setiap kelas pada *class diagram* menjadi tabel. Hal ini merupakan pendekatan yang paling sederhana namun mendatangkan kelemahan ketika rancangan *class diagram* mengalami perubahan. Strategi ini merupakan strategi yang diterapkan pada penelitian ini.
3. Menyatukan *subclass* dan *superclass* pada hirarki yang sama kemudian didefinisikan setiap kelas menjadi tabel. Pendekatan ini menyebabkan cepat berkurangnya kapasitas penyimpanan.

2.9 *Model View Controller* (MVC)

Merupakan pola arsitektur yang membagi aplikasi ke dalam tiga komponen yang saling berinteraksi satu dengan yang lainya (Sommerville, 2011). Setiap komponen pada arsitektur MVC memiliki tugas atau peran yang berbeda.

Komponen *model* memiliki peran untuk menyediakan fungsi logika bisnis atau fungsi untuk mengakses data pada tempat penyimpanan yang akan digunakan oleh komponen *view* untuk ditampilkan kepada pengguna sistem.

Peran komponen *view* adalah untuk menampilkan informasi kepada pengguna. Informasi yang ditampilkan dapat berubah mengikuti perubahan data dan keadaan *model*. Penggunaan model ini memungkinkan penggunaan satu buah *model* untuk memenuhi satu atau beberapa komponen *view.*

Controller pada pola arsitektur MVC berbepran untuk menerima dan mengolah interaksi pengguna yang diterima oleh komponen *view*. C*ontroller* juga mampu berinteraksi dengan komponen *model* untuk mengirimkan permintaan pembaruan data.

Pada implementasinya pola arsitektur MVC banyak digunakan pada kerangka kerja pemrogaman PHP. Kerangka kerja pemrogaman PHP yang digunakan pada penelitian ini adalah Laravel.

2.9.1 Kerangka Kerja Laravel

Laravel adalah kerangka kerja aplikasi web dengan sintaks yang ekspresif dan elegan. Hadirnya laravel membuat proses pengembangan menjadi lebih menyenangkan untuk pengembang tanpa harus mengorbankan fungsionalitas aplikasi yang sedang dikembangkan. Laravel merupakan gabungan dari hal-hal baik yang dipelajari dari kerangka kerja aplikasi web lain. Oleh karena itu, kerangka kerja ini menyediakan kemampuan yang dapat mendukung web berskala besar sekalipun (Laravel, 2014). Hingga sekarang Laravel terus tumbuh dan mengenalkan versi terbaru kepada pengguna. Laravel juga memiliki dokumentasi yang baik yang dikemas dalam Laravel *documentation*, dokumentasi ini sangat lengkap mulai dari versi Laravel 4.2 hingga versi paling baru Laravel 5.5.

Laravel juga mempunyai template sendiri yang disebut dengan *blade* template. *Blade* memudahkan pengguna dalam implementasi berbagai tugas yang sering digunakan seperti: *authentication, session, queue analysis, dan RESTful routing*.

Laravel juga memiliki forum, tempat dimana semua pengguna kerangka kerja ini berkumpul dan berdiskusi membantu pengguna baru yang masih awam tentang Laravel. Forum ini bernama *Laracast*.

2.10 Pengujian Perangkat Lunak

Pengujian perangkat lunak dilakukan untuk menunjukan bahwa sebuah perangkat lunak yang telah dikembangkan dapat berjalan sesuai dengan tujuan awal. Pengujian juga memiliki manfaat untuk mengetahui kekurangan perangkat lunak (Sommerville, 2011). Secara garis besar pengujian perangkat lunak bertujuan dua hal, yaitu:

1. Mengetahui jika perangkat lunak tidak berjalan sesuai dengan yang diharapkan atau tidak memenuhi spesifikasi awal.
2. Menunjukan kepada pelanggan dan pengembang bahwa perangkat lunak telah memenuhi persyaratan.

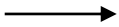
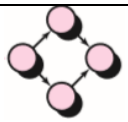
Ada tiga buah pengujian perangkat lunak yang digunakan pada penelitian ini, yaitu pengujian *basis path,* pengujian validasi, dan pengujian kompatibilitas.

2.10.1 Pengujian *Basis Path*

Merupakan salah satu metode dari pengujian *white box* yang diusulkan pertama kali oleh Tom McCabe (1996). Metode ini mengukur kompleksitas dari jalur logis dan menggunakanya sebagai pedoman menetapkan jalur eksekusi. Dari jalur tersebut dapat dilakukan uji coba pada suatu *test case*.

Notasi yang digunakan untuk menggambarkan jalur eksekusi adalah notasi diagram alir, dimana menggunakan notasi lingkaran untuk simpul atau node, dan anak panah untuk *link* atau *edge*. Notasi ini menggambarkan aliran logika yang digunakan dalam suatu bahasa pemrogaman. Tabel 2.15 menunjukan contoh notasi yang digunakan dalam menggambarkan suatu diagram alir.

Tabel 2.15 Notasi *flow graph*

Notasi	Nama	Deskripsi
	*Node*	Notasi yang memvisualisasikan satu atau lebih perintah prosedural pada logika pemrogaman.
	*Edge*	Notasi yang menggambarkan aliran kontrol dari logika pemrogaman.
	*Region*	Merupakan istilah untuk menunjukan daerah yang dibatasi oleh *edge* dan *node,* termasuk juga daerah diluar diagram alir.

Sumber: Diadaptasi dari Pressman (2010)

Cyclomatic Complexicty merupakan ukuran yang menunjukan kompleksitas logika suatu pemrogaman, dapat dihitung menggunakan persamaan V(G) = E-N+2. McCabe (1996) menjelaskan, tingkat kompleksitas pada suatu jalur logis untuk

dapat dengan mudah dikelola dan dimengerti harus bernilai antara 10 hingga 15. Jika nilai kompleksitas lebih dari 15 maka jalur logis tidak bisa diterima dan disarankan untuk melakukan pemecahan jalur logis menjadi lebih sederhana.

2.10.2 Pengujian Validasi

Merupakan pengujian yang berforkus pada hasil dari sistem yang bisa dilihat oleh pengguna (Pressman, 2010). Metode pengujian validasi dilakukan melalui serangkaian kegiatan untuk menunjukan bahwa perangkat lunak telah memenuhi persyaratan. Pengujian validasi menggunakan alur *use case* sebagai panduan pengujian. *Use case* menyediakan skenario yang memungkinkan bagi peneliti untuk mendeteksi kesalahan pada perangkat lunak. Setelah dilakukan pengujian validasi akan diperoleh dua buah hasil, yaitu:

1. Pengetahuan tingkat penerimaan karakteristik fungsi atau performa sistem oleh pelanggan atau pemangku kepentingan.
2. Kesalahan spesifikasi akan terlihat dan kekurangan perangkat lunak akan dicatat.

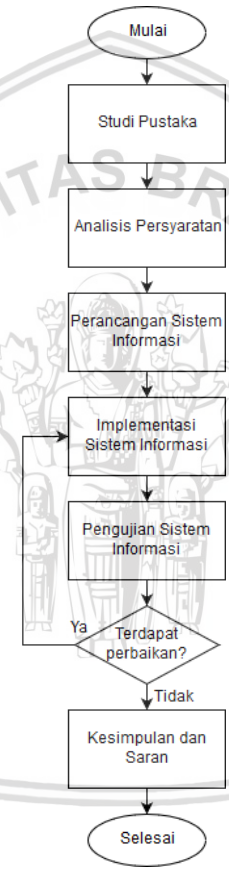
2.10.3 Pengujian Kompatibilitas

Pengujian kompatibilitas merupakan pengujian yang dilakukan untuk mengetahui apakah aplikasi memiliki masalah ketika berjalan pada kecepatan jaringan, aplikasi peramban, sistem operasi, perangkat tampilan, atau komputer yang berbeda (Pressman, 2010). Pengujian kompatibilitas pada aplikasi berbasis web dilakukan dengan mendefinisikan sekumpulan konfigurasi komputasi pada perangkat klien dan menguji aplikasi pada konfigurasi tersebut. Konfigurasi dapat berupa penggunaan aplikasi peramban yang berbeda, sistem operasi berbeda, atau kecepatan jaringan yang berbeda. Dapat disimpulkan pengujian kompatibilitas dilakukan untuk mengetahui kesalahan atau masalah pada eksekusi aplikasi di dalam beberapa konfigurasi yang berbeda.

BAB 3 METODOLOGI

Bab ini menjelaskan alur penelitian yang dilakukan oleh peneliti selama melakukan penelitian. Alur yang dilakukan pada penelitian ini di antaranya adalah studi pustaka, analisis persyaratan, perancangan sistem informasi, implementasi sistem informasi, pengujian sistem informasi, dan terakhir penarikan kesimpulan dan pemberian saran. Diagram alur penelitian dapat ditunjukan dalam Gambar 3.1.

3.1 Alur Penelitian



Gambar 3.1 Alur penelitian

3.2 Studi Pustaka

Penelitian ini dimulai dengan melakukan studi pustaka, yaitu dengan mempelajari beberapa penelitian terdahulu mengenai penggunaan pendekatan *object oriented* dalam pengembangan sistem, penggunaan UML dalam pemodelan sistem, implementasi sistem menggunakan pendekatan *object oriented*, dan teori mengenai pengujian perangkat lunak. Secara khusus, studi pustaka dilakukan guna mempelajari teori yang berkaitan dan digunakan sebagai panduan untuk menyelesaikan penelitian, yaitu teori mengenai proses bisnis dan pemodelanya,

teknik pemodelan menggunakan *use case diagram, activity diagram, sequence diagram, class diagram,* perancangan basis data, pemrogaman php menggunakan kerangka kerja Laravel, dan teori pengujian perangkat lunak menggunakan metode *basis path*, pengujian validasi, dan pengujian *compatibility*.

3.3 Analisis Persyaratan

Tahap analisis persyaratan dimulai dengan melakukan kegiatan observasi dan wawancara untuk mendapatkan informasi mengenai penerapan proses bisnis organisasi dalam pembuatan dokumentasi pekerjaan dan penggajian pegawai. Selain itu, tahap ini juga dilakukan untuk mengumpulkan beberapa dokumen yang berkaitan dengan kebutuhan penelitian. Beberapa dokumen yang dikumpulkan adalah contoh dokumentasi pekerjaan, foto pekerjaan, data pegawai, contoh lembar kehadiran pegawai, dan contoh dokumen perhitungan keuangan pegawai.

Kegiatan selanjutnya adalah memodelkan dan menganalisis proses bisnis yang berjalan pada perusahaan dan menentukan proses bisnis usulan dari peneliti. Kemudian dilakukan identifikasi pemangku kepentingan untuk mendapatkan informasi mengenai pemangku kepentingan dan tipe-tipe pemangku kepentingan pada lokasi studi kasus penelitian.

Kemudian dilakukan analisis masalah terhadap hasil wawancara dari pemangku kepentingan. Hasil analisis masalah kemudian didokumentasikan ke dalam tabulasi pernyataan masalah. Tabulasi pernyataan masalah juga mendokumentasikan solusi yang diusulkan untuk menyelesaikan masalah yang telah teridentifikasi.

Kegiatan berikutnya ialah identifikasi kebutuhan pemangku kepentingan dan pengguna. Kegiatan ini dilakukan untuk mengetahui berbagai macam kebutuhan pengguna, situasi yang terjadi saat ini untuk memenuhi kebutuhan pengguna, dan beberapa solusi yang diusulkan untuk memenuhi kebutuhan pengguna.

Selanjutnya identifikasi pengguna dilakukan untuk mengetahui siapa saja calon pengguna sitem. Identifikasi fitur juga dilakukan untuk mengetahui fitur sistem informasi yang akan dikembangkan sebagai bentuk solusi yang akan disediakan oleh sistem bagi pengguna atau pemangku kepentingan untuk memenuhi kebutuhanya.

Kemudian, pemodelan *use case* dilakukan untuk mengetahui tujuan aktor dalam menggunakan sistem informasi yang akan dikembangkan. Pemodelan *use case* juga menunjukan bagaimana alur penggunaan sistem yang dilakukan oleh aktor. Selanjutnya, alur penggunaan sistem pada spesifikasi *use case* digunakan sebagai panduan pemodelan aktivitas ke dalam *activity diagram*.

Tahap analisis persyaratan akan dilakukan kembali apabila hasil kegiatan validasi sistem informasi dari tahap pengujian menunjukan bahwa sistem informasi yang dikembangkan belum memenuhi kebutuhan pengguna.

3.4 Perancangan Sistem Informasi

Perancangan sistem informasi dilakukan untuk memodelkan sistem informasi ke dalam notasi pada UML berdasarkan hasil dari tahap analisis persyaratan. Perancangan sistem berisi hasil pemodelan interaksi objek ke dalam *sequence diagram,* pemodelan *class diagram* secara umum yang diimplementasikan ke dalam pemodelan *class diagram* kerangka kerja Laravel sebagai acuan implementasi sistem informasi menggunakan kerangka kerja Laravel, pemodelan *class diagram domain model* yang menjadi acuan dari perancangan basis data ke dalam bentuk PDM, perancangan algoritme dalam bentuk *pseducode*, dan perancangan antarmuka pengguna.

3.5 Implementasi Sistem Informasi

Implementasi sistem informasi merupakan suatu tahap pembuatan sistem informasi berdasarkan hasil perancangan sistem. Sistem informasi dikembangkan menggunakan bahasa pemrogaman PHP dan Javascript, memanfaatkan kerangka kerja Laravel. Hasil tahap pembuatan sistem informasi adalah sebuah prototipe sistem berbasis web yang mendemonstrasikan fungsi-fungsi sesuai dengan yang telah teridentifikasi pada tahap analisis persyaratan.

Kegiatan dokumentasi juga dilakukan dalam tahap implementasi. Dokumentasi tahap implementasi berisi informasi mengenai spesifikasi lingkungan perangkat keras, spesifikasi lingkungan perangkat lunak, dan spesifikasi lingkungan *deployment*. Selain itu, didokumentasikan pula hasil implementasi algoritme dalam bentuk kode program dan antarmuka pengguna.

3.6 Pengujian Sistem Informasi

Tahap pengujian sistem informasi dilakukan untuk memastikan bahwa sistem informasi yang telah dikembangkan dapat berjalan sesuai dengan kebutuhan dan harapan pengguna. Kegiatan pengujian dilakukan dengan menerapkan teknik pengujian *basis path*, pengujian validasi, dan pengujian kompatibilitas.

Pengujian *basis path* dilakukan mengacu pada skenario yang didapatkan dari hasil perhitungan *cyclomatic complexity* terhadap *flow graph*. Hasil dari menjalankan skenario akan dibandingkan dengan hasil yang diharapkan oleh pengguna.

Pengujian validasi dilakukan berdasarkan skenario penggunaan sistem oleh aktor yang didapatkan dari alur *use case diagram* pada spesifikasi *use case diagram*. Hasil yang diperoleh dari skenario selanjutnya dibandingkan dengan hasil yang diharapkan oleh pengguna.

Pengujian kompatibilitas dilakukan dengan menggunakan alat bantu perangkat lunak SortSite untuk menguji kompatibilitas sistem informasi terhadap beberapa aplikasi peramban web berbeda. Jika diketahui masih terdapat hasil pengujian yang

menyatakan bahwa sistem membutuhkan perbaikan, maka dilakukan kegiatan perbaikan pada tahap implementasi.

3.7 Kesimpulan dan Saran

Pembuatan kesimpulan merupakan tahap akhir dari penelitian untuk menentukan apakah hasil penelitian dapat menjawab rumusan masalah. Pada tahap ini juga menjelaskan saran untuk pengembangan atau penelitian di masa yang akan datang.

BAB 4 PEMODELAN PROSES BISNIS & ANALISIS PERSYARATAN SISTEM

Bab ini menjelaskan pemodelan prosesn bisnis dan analisis persayaratan yang diperoleh berdasarkan analisis terhadap hasil wawancara dan observasi. Pada bab ini ditunjukan hasil analisis yang terdiri dari model proses bisnis, tipe pemangku kebutuhan, pernyataan masalah, kebutuhan pengguna, pengguna sistem informasi, fitur sistem informasi, persyaratan fungsional dan nonfungsional, *use case,* dan *activity diagram*.

4.1 Pemodelan Proses Bisnis

Pemodelan proses bisnis dilakukan untuk memodelkan serangkaian aktivitas yang dilakukan oleh perusahaan untuk memenuhi kebutuhan pengguna bisnis. Penggalian informasi mengenai proses bisnis yang dimodelkan diperoleh dengan cara melakukan wawancara kepada petugas admin dan observasi mengenai alur proses bisnis. Dalam penelitian ini, pemodelan proses bisnis akan menghasilkan dua kelompok pemodelan proses bisnis, yaitu proses bisnis aktual berjalan (*as-is*) dan proses bisnis usulan (*to-be*).

4.2 Proses Bisnis *As-Is*

Pada penelitian ini proses bisnis *as-is* dapat diidentifikasi berdasarkan keadaan proses bisnis yang berjalan pada lokasi studi kasus sebelum solusi baru diusulkan. Analisis proses bisnis *as-is* dilakukan dengan melakukan wawancara kepada pegawai perusahaan dan observasi mengenai alur proses bisnis. Kemudian analisis hasil wawancara dimodelkan dengan notasi BPMN untuk membantu pemangku kepentingan memahami hasil analisis proses bisnis *as-is*. Pemodelan dan analisis pada proses bisnis *as-is* dilakukan supaya peneliti dapat mengidentifikasi kemungkinan peningkatan atau perbaikan terhadap proses bisnis melalui solusi yang akan ditawarkan dalam penelitian ini.

4.2.1 Proses Bisnis *As-Is* Pembuatan Dokumentasi Pekerjaan

Pada Gambar 4.1 merupakan rangkaian aktivitas pembuatan dokumentasi pekerjaan yang dilakukan oleh fungsi bisnis staf petugas lapangan, petugas admin, dan petugas admin cabang lain untuk pekerjaan yang area kerjanya meliputi banyak cabang. Proses bisnis dimulai pada saat petugas lapangan menyerahkan foto dan lembar pekerjaan kepada petugas admin di area kerjanya, foto dan lembar pekerjaan yang telah diberikan selanjutnya dicatat dan disimpan ke dalam suatu dokumen dengan format .xls. Ketika datang jadwal pembuatan dokumentasi jika pekerjaan meliputi banyak cabang maka dokumen hasil pencatatan tersebut akan dikumpulkan dari setiap cabang dan disusun menjadi dokumentasi pekerjaan yang utuh. Kemudian petugas admin akan mencetak dokumentasi pekerjaan yang telah tersusun.

4.2.2 Proses Bisnis *As-Is* Pembuatan Laporan Keuangan (Bagian Pencatatan)

Proses bisnis *as-is* pembuatan laporan keuangan pada Gambar 4.2 merupakan serangkaian aktivitas yang dilakukan oleh fungsi bisnis staf petugas admin atau beserta staf petugas admin cabang lain jika pekerjaan meliputi banyak cabang. Proses bisnis dimulai pada saat petugas admin mendapatkan lembar kehadiran, nota pengeluaran, atau bukti-bukti keuangan lain dari petugas lapangan. Kemudian mencatat data kehadiran dan keuangan pegawai, dari pencatatan tersebut dihasilkan suatu catatan kehadiran dan keuangan pegawai. Jika pekerjaan meliputi banyak cabang maka catatan tersebut selanjutnya dikumpulkan dari seluruh cabang dengan pekerjaan serupa.

4.2.3 Proses Bisnis *As-Is* Pembuatan Laporan Keuangan (Bagian Perhitungan)

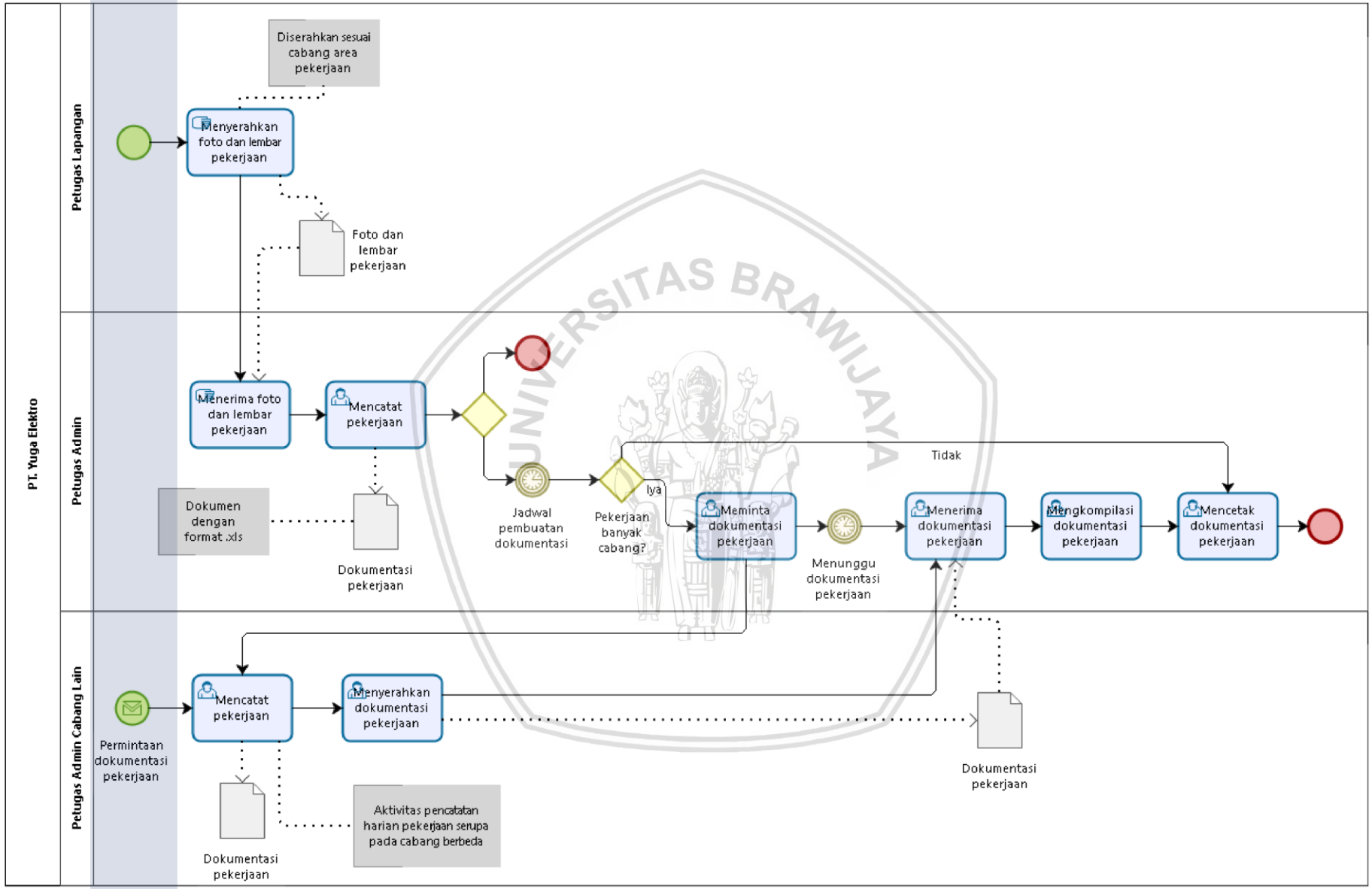
Pada Gambar 4.3 merupakan rangkaian aktivitas yang dilakukan oleh fungsi bisnis staf petugas admin. Proses bisnis dilakukan setelah proses bisnis pencatatan selesai dilakukan dan tiba waktu untuk perhitungan keuangan. Proses bisnis dimulai pada saat petugas admin menghitung setiap pendapatan pegawai berdasarkan dokumen catatan kehadiran dan keuangan. Selanjutnya akan disusun laporan keuangan dan mencetak laporan tersebut.

4.3 Pemodelan Proses Bisnis To-Be

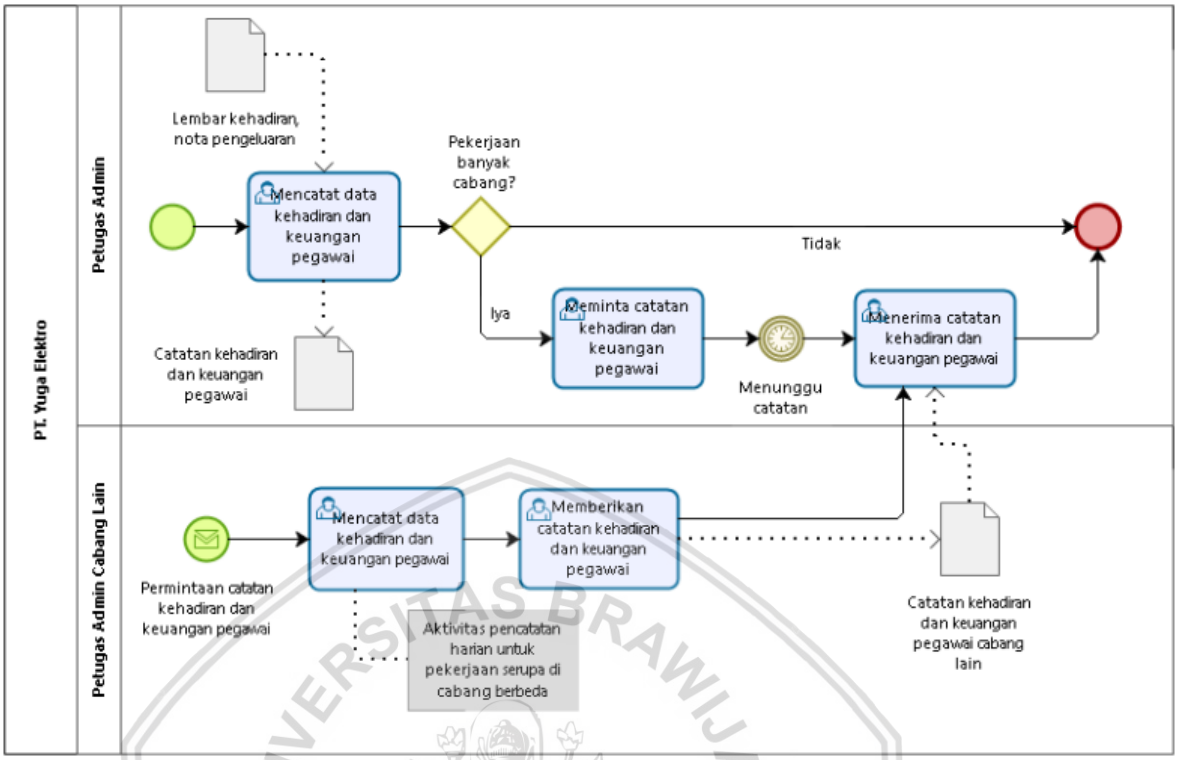
Proses bisnis*to-be* yang dimodelkan pada penelitian ini merupakan proses bisnis usulan yang ditawarkan kepada pemangku kepentingan. Proses bisnis *to-be* diidentifikasi berdasarakan hasil analisis terhadap proses bisnis *as-is*. Dalam penelitian ini, analisis proses bisnis *to-be* dilakukan dengan melakukan eliminasi pada aktivitas yang dapat dipangkas atau dihilangkan waktu pengerjaanya dengan menggunakan dukungan sistem informasi. Selanjutnya, proses bisnis *to-be* dimodelkan untuk menggambarkan perubahan atau tambahan pada proses bisnis yang diusulkan melalui penelitian ini. Hasil proses bisnis *to-be* digunakan sebagai masukan kegiatan analisis persyaratan.

4.3.1 Proses Bisnis *To-Be* Pembuatan Dokumentasi Pekerjaan

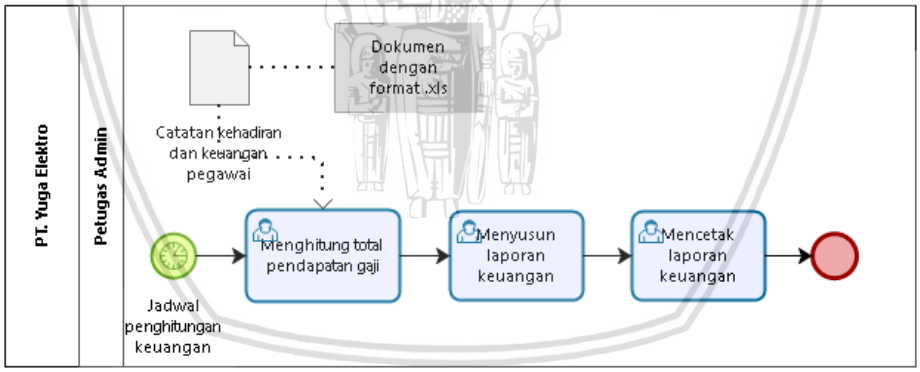
Proses bisnis to-be pembuatan dokumentasi pekerjaan dalam Gambar 4.4 merupakan proses bisnis usulan yang memuat visualisasi solusi dengan menggunakan lokasi penyimpanan data terpusat untuk menyimpan informasi data pekerjaan. Perubahan aktvitas pada proses bisnis pembuatan dokumentasi pekerjaan dijelaskan pada Tabel 4.1



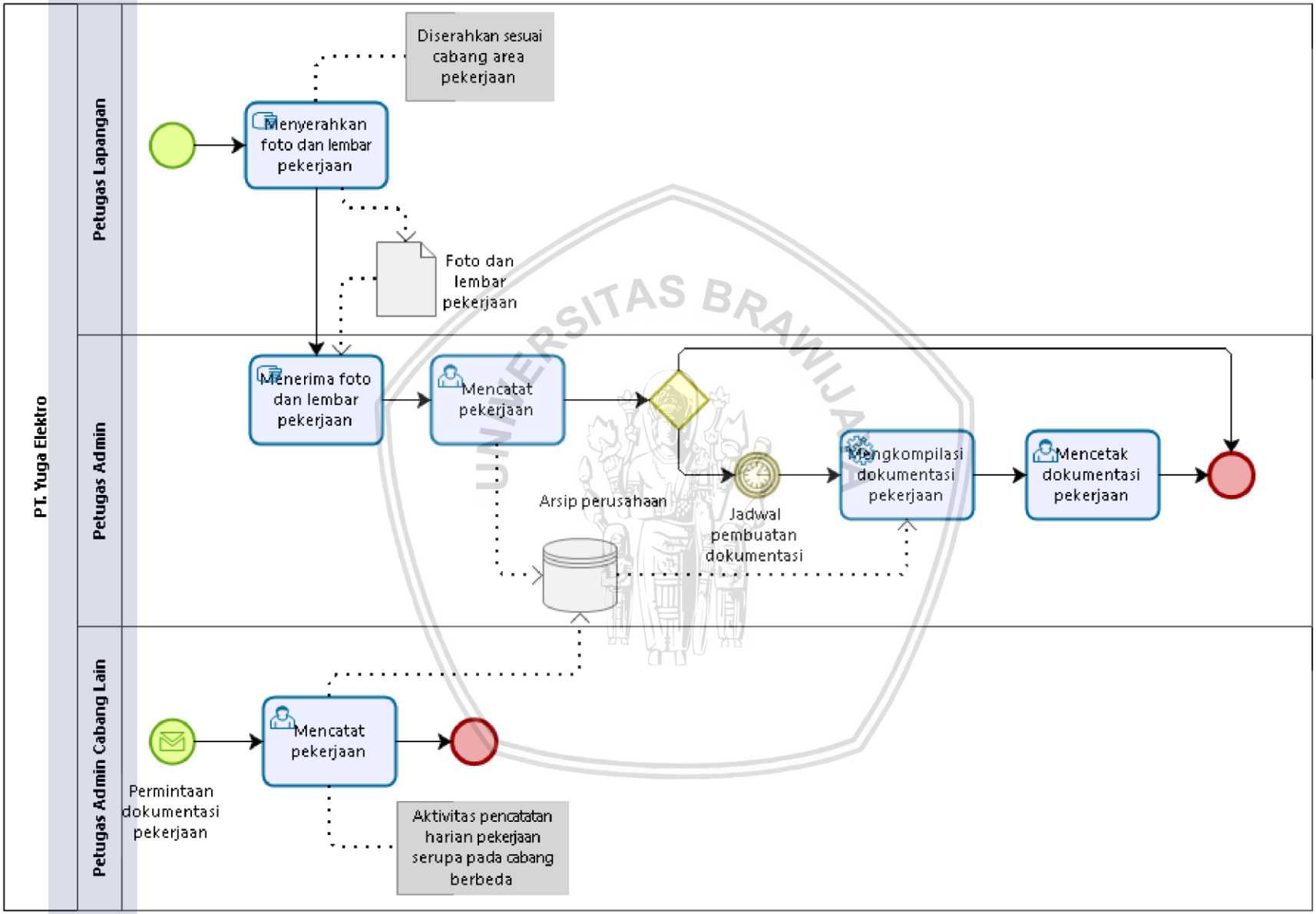
Gambar 4.1 Proses bisnis As-Is pada pembuatan dokumentasi pekerjaan



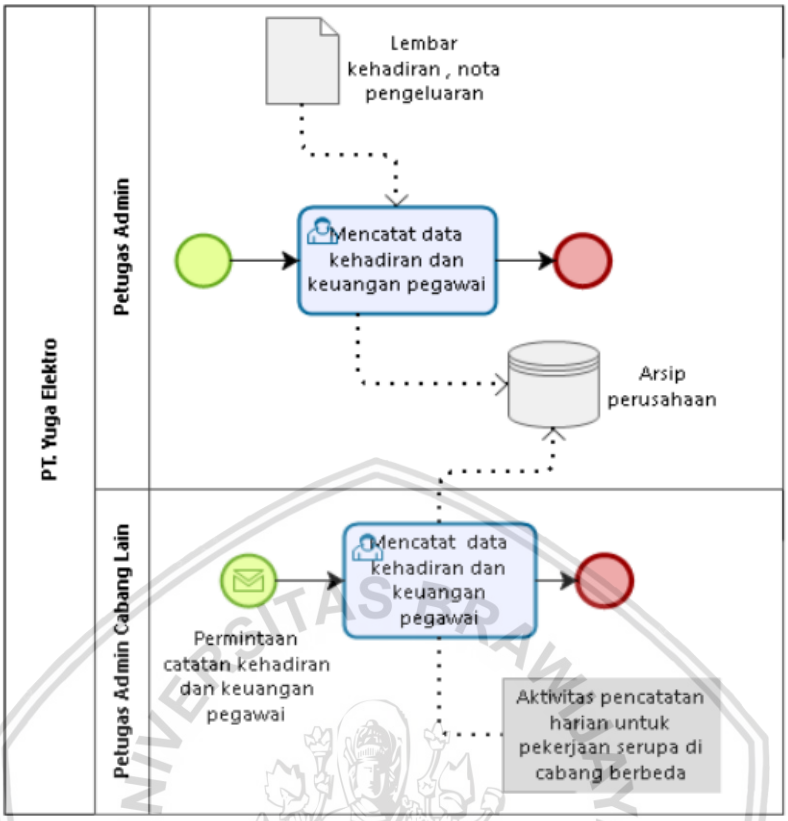
Gambar 4.2 Proses bisnis As-Is pada pembuatan laporan keuangan (bagian pencatatan)



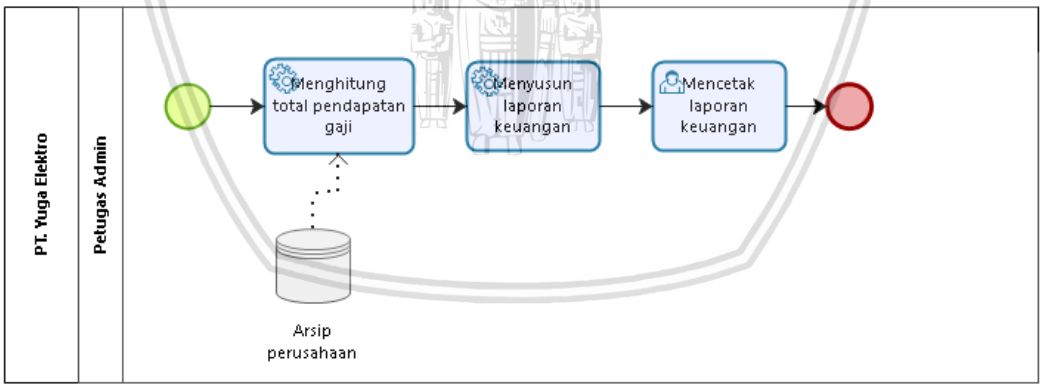
Gambar 4.3 Proses bisnis As-Is pada pembuatan laporan keuangan (bagian perhitungan)



Gambar 4.4 Proses bisnis To-Be pada pembuatan dokumentasi pekerjaan



Gambar 4.5 Proses bisnis To-Be pada pembuatan laporan keuangan (bagian pencatatan)



Gambar 4.6 Proses bisnis To-Be pada pembuatan laporan keuangan (bagian perhitungan)

Tabel 4.1 Perubahan aktivitas proses bisnis pembuatan dokumentasi pekerjaan

Unit Bisnis	Proses Bisnis *As-Is*	Proses Bisnis *To-Be*	Keterangan
Petugas Admin	Meminta dokumentasi pekerjaan	-	Dieliminasi
Petugas Admin	Menerima dokumentasi pekerjaan	-	Dieliminasi
Petugas Admin Cabang Lain	Menyerahkan dokumentasi pekerjaan	-	Dieliminasi
Petugas Admin	Mengkompilasi dokumentasi pekerjaan	-	Diotomatisasi

4.3.2 Proses Bisnis *To-Be* Pembuatan Laporan Keuangan

Proses bisnis *to-be* pembuatan laporan keuangan yang ditunjukan oleh Gambar 4.5 dan Gambar 4.6 merupakan proses bisnis usulan terkait kegiatan pembuatan laporan keuangan. Solusi yang ditawarkan untuk proses pembuatan laporan keuangan adalah penggunaan lokasi penyimpanan data terpusat untuk kebutuhan pengelolaan data keuangan dan perhitungan secara otomatis dan *real time*. Perubahan aktivitas proses bisnis pembuatan laporan keuangan ditunjukan pada Tabel 4.2.

Tabel 4.2 Perubahan aktivitas proses bisnis pembuatan laporan keuangan

Unit Bisnis	Proses Bisnis *As-Is*	Proses Bisnis *To-Be*	Keterangan
Petugas Admin	Meminta catatan kehadiran dan keuangan pegawai	-	Dieliminasi
Petugas Admin	Menerima catatan kehadiran dan keuangan pegawai	-	Dieliminasi
Petugas Admin Cabang Lain	Memberikan catatan kehadiran dan keuangan pegawai	-	Dieliminasi
Petugas Admin	Menghitung total pendapatan gaji	-	Diotomatisasi
Petugas Admin	Menyusun laporan keuangan	-	Diotomatisasi

4.4 Analisis Persayaratan

4.4.1 Identifikasi Tipe Pemangku Kepentingan

Identifiksai tipe pemangku kepentingan merupakan aktivitas yang dilakukan untuk mengetahui dan mengelompokan beberapa pemangku kepentingan yang terlibat pada penelitian ini sesuai dengan karakteristik pemangku kepentingan dan hubungan antara pemangku kepentingan dengan sistem yang akan dikembangkan. Informasi pemangku kepentingan diperoleh dengan melakukan analisis terhadap hasil wawancara dengan petugas admin perusahaan.

Hasil identifikasi tipe pemangku kepentingan adalah daftar contoh pemangku kepentingan yang telah di kelompokan sesuai karakteristik dan hubungan pemangku kepentingan terhadap sistem yang akan dikembangkan. Hasil identifikasi pemangku kepentingan digunakan sebagai informasi untuk kegiatan analisis permasalahan, sehingga dapat diperoleh informasi masalah dari berbagai pemangku kepentingan yang berada pada lokasi studi kasus. Penjelasan tipe pemangku kepentingan terdapat pada Tabel 4.3.

Tabel 4.3 Tipe pemangku kepentingan

Tipe Pemangku Kepentingan	Deskripsi	Pemangku Kepentingan
Pengguna	Merupakan Individu yang berinteraksi dengan sistem informasi secara langsung dan berperan sebagai aktor pada *use case*.	• Petugas Admin • Direktur
Pengembang	Organisasi atau individu yang melakukan pengembangan sistem informasi.	Peneliti
Pihak yang berwenang	Organisasi atau individu yang memiliki kemampuan untuk mendukung pengembangan solusi dengan cara memberikan informasi mengenai regulasi yang berjalan, sehingga solusi yang akan dikembangkan dapat berjalan sesuai dengan aturan yang berlaku	PT. Yuga Elektro Tuban
Pelanggan	Organisasi atau individu yang akan mendapatkan manfaat dari sistem informasi yang dikembangkan.	• PT. Yuga Elektro Tuban • Petugas Lapangan

4.4.2 Analisis Masalah

Analisis masalah dilakukan untuk memahami masalah yang terjadi dan harus terselesaikan oleh pemangku kepentingan. Dalam penelitian ini, analisis masalah dilakukan dengan melakukan analisis hasil wawancara terhadap petugas admin.

Problem statement yang didapatkan dari analisis hasil wawancara terdapat pada Tabel 4.4.

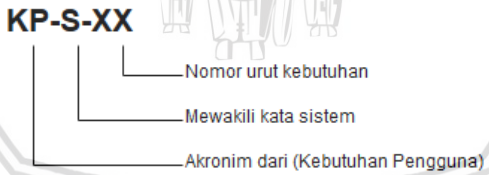
Tabel 4.4 *Problem statement*

Masalah	• Penyimpanan data yang tidak efisien di sebabkan oleh penyimpanan data yang masih dilakukan di tiap cabang area pekerjaan.
Mempengaruhi	Pengguna, pihak yang berwenang, pelanggan.
Dampak	• Proses pengumpulan data dari berbagai cabang menyulitkan pekerjaan dari petugas admin.
Solusi	Sistem yang menyediakan layanan pengelolaan data terintegrasi dalam lingkup perusahaan.

4.4.3 Identifikasi Kebutuhan Pemangku Kepentingan dan Pengguna

Informasi mengenai kebutuhan pengguna diperoleh dengan melakukan analisis terhadap *problem statement* dan hasil wawancara dengan petugas admin. Identifikasi kebutuhan pengguna berisi sekumpulan pernyataan yang berhubungan dengan masalah yang telah diidentifikasi pada tahap analisis masalah.

Hasil identifikasi kebutuhan pengguna digunakan sebagai dasar informasi tahap analisis fitur yang merupakan tahap mengidentifikasi solusi yang ditawarkan untuk memenuhi kebutuhan pengguna. Masing-masing kebutuhan pengguna diberikan kode sebagai identitas, keterangan kodifikasi kebutuhan pengguna terdapat pada Gambar 4.7.



Gambar 4.7 Keterangan kodifikasi kebutuhan pengguna

Pada Tabel 4.5 dapat ditunjukan hasil identifikasi kebutuhan pengguna yang berisi informasi kode identitas untuk masing-masing kebutuhan pengguna, pernyataan kebutuhan pengguna, pemangku kepentingan yang termasuk dalam kategori pengguna dan pengembang yang memiliki keterkaitan dengan kebutuhan yang telah diidentifikasi, situasi saat ini yang berkaitan dengan kegiatan yang dilakukan untuk memenuhi kebutuhan pengguna, dan solusi yang ditawarkan untuk memenuhi masing-masing kebutuhan yang telah teridentfikasi.

Tabel 4.5 Hasil identifikasi kebutuhan pengguna

Kode Kebutuhan	Kebutuhan Pengguna	Pemangku Kepentingan	Situasi Saat Ini	Solusi Yang Ditawarkan
KP-S-1	Informasi di dalam sistem harus dapat diakses sesuai dengan identitas pengguna sistem	Petugas admin, direktur	Tidak ada	Sistem informasi menyediakan layanan untuk membatasi akses terhadap informasi tertentu sesuai dengan hak akses yang dimiliki oleh pengguna.
KP-S-2	Sistem harus menyediakan layanan pencatatan pekerjaan	Petugas admin, direktur	Pencatatan pekerjaan dilakukan pada *white board* dan pada dokumen dengan format aplikasi Ms Excel	Sistem informasi yang menyediakan layanan pencatatan pekerjaan yang tersimpan di dalam sebuah basis data yang dapat diakses melalui sistem informasi.
KP-S-3	Sistem harus menyediakan layanan pencatatan pegawai	Petugas admin, direktur	Data pegawai disimpan di dalam format dokumen aplikasi Micorsoft excel	Sistem informasi yang menyediakan layanan pencatatan data pegawai dan disimpan di dalam basis data.
KP-S-4	Sistem harus menyediakan layanan yang dapat membantu pecatatan dan pencetakan laporan pelaksanaan pekerjaan	Petugas admin, direktur	Laporan pelaksanaan pekerjaan disimpan ke dalam format aplikasi Ms Excel.	Sistem informasi yang menyediakan layanan pencatatan pekerjaan yang merekam informasi mengenai data pekerjaan yang telah selesai dikerjakan.

				Setelah informasi pekerjaan tersimpan di dalam sistem, informasi dapat dicetak.
KP-S-5	Sistem harus menyediakan layanan yang dapat membantu pencatatan kehadiran pegawai	Petugas admin, direktur	Data kehadiran tersimpan di dalam dokumen cetak dan di letakan pada ruang arsip perusahaan. Data kehadiran diisi setiap pagi hari sebelum pegawai melaksanakan pekerjaan.	Sistem informasi yang menyediakan pencatatan data kehadiran dan disimpan di dalam sebuah basis data yang dapat diakses melalui sistem informasi.
KP-S-6	Sistem harus menyediakan layanan yang dapat membantu proses perhitungan keuangan.	Petugas admin, direktur	Perhitungan dilakukan menggunakan bantuan aplikasi Micorsoft excel dengan mengacu pada kehadiran dan bukti cetak keuangan lainya.	Sistem informasi yang secara otomatis menghitung keuangan pada setiap bulan.
KP-S-7	Sistem harus menyediakan layanan yang membantu penyediaan dan pencetakan laporan keuangan.	Petugas admin, direktur	Laporan keuangan dibuat secara manual dengan meringkas hasil perhitungan dan hasilnya disimpan dalam format aplikasi Ms Excel.	Sistem informasi yang menyediakan layanan untuk menyediakan laporan keuangan dan setelah tersimpan dapat dicetak.
KP-S-8	Sistem harus menyediakan layanan pencatatan aktivitas yang dilakukan sebagai bentuk pertanggung	Petugas admin, direktur	Tidak ada	Sistem informasi menyediakan layanan yang menyimpan segala bentuk aktivitas kegiatan yang dilakukan

	jawaban dalam bekerja.			terhadap sistem dan disimpan di dalam sebuah basis data.
KP-S-9	Sistem harus menyediakan layanan untuk mengelola data yang harus dimuat sebelum sistem dapat digunakan untuk pertama kali oleh pengguna.	Direktur	Tidak ada	Sistem informasi menyediakan layanan untuk pemuatan data awal pengguna.

4.4.4 Identifikasi Pengguna

Identifikasi pengguna merupakan tahap yang dilakukan untuk mengetahui individu yang akan menggunakan sistem informasi. Pengguna sistem merupakan bagian dari kelompok kategori pengguna dan pengembang dalam tipe pemangku kepentingan.

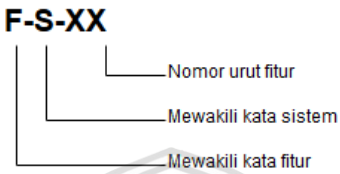
Hasil identifikasi pengguna memuat tipe pemangku kepentingan yang tergolong ke dalam kategori pengembang dan pengguna, tipe pengguna, dan deskripsi pengguna sistem informasi. Informasi mengenai idenfikasi pengguna digunakan untuk melakukan identifikasi aktor yang akan dimodelkan ke dalam diagram *use case.* Penjelasan hasil identifikasi pengguna terdapat pada Tabel 4.6.

Tabel 4.6 Hasil identifikasi pengguna

Tipe Pemangku Kepentingan	Tipe Pengguna	Deskripsi
Pengembang	Peneliti	Pengguna yang memiliki semua akses sistem informasi dan mengelola data awal sistem informasi sebelum dapat digunakan.
Pengguna	Petugas Admin	Petugas perusahaan yang bertugas untuk mengelola pekerjaan, mengelola pegawai, mengelola dokumentasi, mengelola kehadiran, dan mengelola keuangan. Petugas admin memiliki akses terbatas terhadap sistem.
	Direktur	Pengawas jalanya proses administrasi pekerjaan dan memiliki kedudukan tertinggi pada struktur organisasi perusahaan. Direktur memiliki semua akses terhadap sistem informasi.

4.4.5 Identifikasi Fitur

Hasil identifikasi fitur merupakan representasi dari beberapa solusi yang ditawarkan oleh peneliti untuk memenuhi kebutuhan pengguna, sehingga masalah yang dihadapi oleh pemangku kepentingan bisa terselesaikan. Hasil identifikasi fitur ini merupakan deskripsi singkat dari layanan-layanan yang disediakan oleh sistem untuk dapat memenuhi kebutuhan pengguna. Masing-masing fitur pada penelitian ini memiliki kode sebagai identitas yang mengacu pada Gambar 4.8.



Gambar 4.8 Keterangan kodifikasi fitur

Hasil identifikasi fitur akan digunakan sebagai informasi untuk melakukan identifikasi persyaratan fungsional dan nonfungsional sistem informasi yang akan dikembangkan. Informasi mengenai identifikasi fitur dan deskripsi singkatnya dapat dilihat pada Tabel 4.7.

Tabel 4.7 Hasil identifikasi fitur

Kode Fitur	Deskripsi
F-S-1	Sistem dapat digunakan untuk mengenali identitas pengguna sistem dan membatasi akses pengguna terhadap informasi dan layanan yang disesuaikan oleh sistem berdasarkan hak akses pengguna.
F-S-2	Sistem dapat digunakan untuk mengelola pekerjaan, antara lain menambah pekerjaan, mengubah status pekerjaan, dan menghapus pekerjaan.
F-S-3	Sistem dapat digunakan untuk mengelola pegawai, antara lain menambah pegawai, mengubah pegawai, menghapus pegawai, mencetak daftar pegawai, dan alokasi pegawai ke dalam pekerjaan.
F-S-4	Sistem dapat digunakan untuk mengelola dokumentasi, antara lain menambah dokumentasi, mengubah dokumentasi, menghapus dokumentasi, mencetak daftar dokumentasi.
F-S-5	Sistem dapat digunakan untuk mengelola kehadiran antara lain, menambah data kehadiran, batal kehadiran, mencetak data kehadiran.
F-S-6	Sistem dapat digunakan untuk mengelola keuangan antara lain, menambah data keuangan, menghapus data keuangan, mengatur rumus dan mencetak data keuangan.
F-S-7	Sistem dapat digunakan untuk menampilkan riwayat aktivitas pengguna terhadap sistem dan informasi di dalamnya.

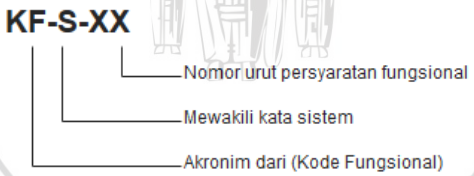
F-S-8	Sistem dapat digunakan untuk mengelola admin, antara lain menambah admin, mengubah data admin, dan menghapus data admin.

Fitur-fitur yang teridentifikasi pada penelitian ini selanjutnya dikelompokan berdasarkan keterkaitanya terhadap kebutuhan pengguna yang harus dipenuhi. Pengelompokan ini menunjukan bahwa sebuah fitur merupakan solusi untuk memenuhi satu atau beberapa kebutuhan pengguna. Pengelompokan fitur dengan kebutuhan pengguna dapat ditunjukan pada Tabel 4.8.

Tabel 4.8 Hubungan kebutuhan pengguna dengan fitur

Kebutuhan Pengguna	Fitur
KP-S-1	F-S-1
KP-S-2	F-S-2
KP-S-3	F-S-3
KP-S-4	F-S-4
KP-S-5	F-S-5
KP-S-6	F-S-6
KP-S-7	
KP-S-8	F-S-7
KP-S-9	F-S-8

4.4.6 Persyaratan Fungsional



Gambar 4.9 Keterangan kodifikasi persyaratan fungsional

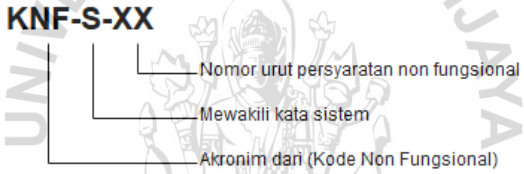
Persyaratan fungsional menjelaskan kondisi atau kemampuan yang harus terpenuhi oleh sistem untuk dapat menjalankan fitur sehingga kebutuhan pengguna dapat terpenuhi. Tabel 4.9 menunjukan hubungan antara fitur dengan persyaratan fungsional pada sistem informasi yang akan dikembangkan. Persyaratan fungsional menggunakan kode sebagai identitas mengacu pada kodifikasi yang terdapat pada Gambar 4.9.

Tabel 4.9 Persyaratan fungsional

Kode Fitur	Kode Persyaratan Fungsional	Deskripsi
F-S-1	KF-S-1.1	Sistem harus menyediakan fungsi untuk membedakan hak akses pada setiap pengguna.
F-S-2	KF-S-2.1	Sistem harus menyediakan fungsi untuk melakukan penambahan data pekerjaan.
	KF-S-2.2	Sistem harus menyediakan fungsi untuk menghapus data pekerjaan.
	KF-S-2.3	Sistem harus menyediakan fungsi untuk mengganti status pekerjaan dikerjakan atau telah selesai.
F-S-3	KF-S-3.1	Sistem harus menyediakan fungsi untuk menambahkan data pegawai baru.
	KF-S-3.2	Sistem harus menyediakan fungsi untuk mengubah data pegawai.
	KF-S-3.3	Sistem harus menyediakan fungsi untuk menghapus pegawai.
	KF-S-3.4	Sistem harus menyediakan fungsi untuk mencetak daftar pegawai.
	KF-S-3-5	Sistem harus menyediakan fungsi untuk melakukan alokasi pegawai ke dalam pekerjaan
F-S-4	KF-S-4.1	Sistem harus menyediakan fungsi untuk menambahkan dokumentasi pekerjaan.
	KF-S-4.2	Sistem harus menyediakan fungsi untuk mengubah data dokumentasi pekerjaan.
	KF-S-4.3	Sistem harus menyediakan fungsi untuk menghapus dokumentasi pekerjaan yang tidak diperlukan kembali.
	KF-S-4.4	Sistem harus menyediakan fungsi untuk mencetak data dokumentasi pekerjaan.
F-S-5	KF-S-5.1	Sistem harus menyediakan fungsi untuk menyimpan kehadiran pegawai.
	KF-S-5.2	Sistem harus menyediakan fungsi untuk membatalkan kehadiran pegawai.
	KF-S-5.3	Sistem harus menyediakan fungsi untuk mencetak rekap kehadiran pegawai.
F-S-6	KF-S-6.1	Sistem harus menyediakan fungsi untuk menambah data keuangan diluar gaji pokok pada bulan tertentu.
	KF-S-6.2	Sistem harus menyediakan fungsi untuk menghapus data keuangan.

	KF-S-6.3	Sistem harus menyediakan fungsi untuk mencetak hasil perhitungan keuangan.
	KF-S-6.4	Sistem harus menyediakan fungsi untuk mengatur rumus keuangan pekerjaan.
	KF-S-6.5	Sistem harus menyediakan fungsi untuk menampilkan data keuangan.
F-S-7	KF-S-7.1	Sistem harus menyediakan fungsi untuk melihat riwayat aktivitas penggunaan sistem.
F-S-8	KF-S-8.1	Sistem harus menyediakan fungsi untuk menambahkan pengguna sistem.
	KF-S-8.2	Sistem harus menyediakan fungsi untuk menghapus pengguna sistem.
	KF-S-8.3	Sistem harus menyediakan fungsi untuk mengubah data pengguna sistem.

4.4.7 Persyaratan Nonfungsional



Gambar 4.10 Keterangan kodifikasi persyaratan nonfungsional

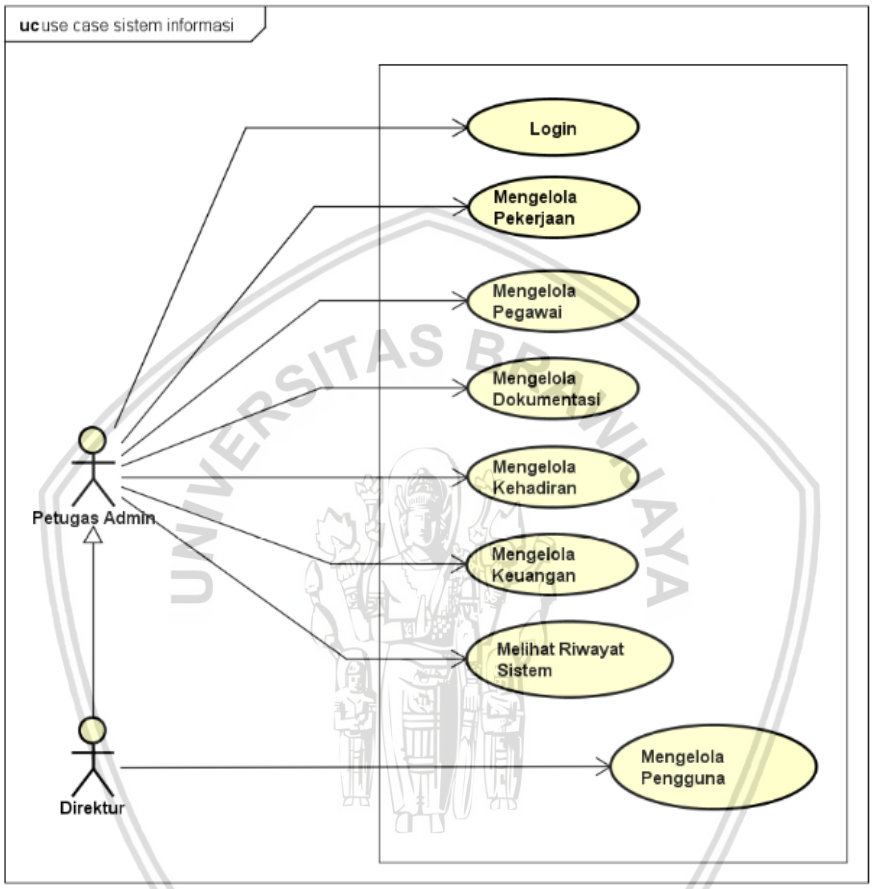
Persyaratan nonfungsional merupakan persyaratan yang harus dipenuhi oleh sistem untuk mendukung berjalanya sistem yang akan digunakan. Mengacu pada kodifikasi pada Gambar 4.10 maka persyaratan nonfungsional dapat ditunjukan sebagaimana pada Tabel 4.10.

Tabel 4.10 Persyaratan nonfungsional

Kode Fitur	Kode Persyaratan NonFungsional	Nama Persyaratan NonFungsional	Deskripsi
F-S-1	KNF-S-1	Kompatibilitas	Sistem informasi dapat berjalan dengan benar dan tanpa kehilangan konten pada minimal tiga aplikasi peramban web yang berbeda.
F-S-2			
F-S-3			
F-S-4			
F-S-5			
F-S-6			
F-S-7			
F-S-8			

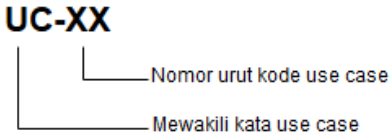
Sebenarnya masih terdapat beberapa persyaratan nonfungsional lain yang bisa diterapkan pada penelitian ini seperti *usability, availability,* performa, ruang penyimpanan, dan keamanan. Namun pada cakupan penelitian ini peneliti hanya akan menguji dan mendokumentasikan persyaratan nonfungsional kompatibilitas.

4.5 Pemodelan *Use Case*



Gambar 4.11 *Use case* sistem informasi

Pemodelan *use case* dilakukan setelah tahap analisis persyaratan. Teknik *use case* yang digunakan pada penelitian ini mengikuti *RUP style* dengan mengacu pada pustaka Bitter, K. & Spence, I (2002). Setiap *use case* memiliki kode sebagai identitas yang mengikuti kodifikasi pada Gambar 4.12.



Gambar 4.12 Keterangan kodifikasi *use case*

Pada pemodelan *use case* dilakukan identifikasi aktor, identifikasi *use case*, dan spesifikasi untuk setiap *use case*. Identifikasi aktor dilakukan dengan mengelompokan pengguna sistem berdasarkan karakteristik yang dimiliki.

Identifikasi *use case* dilakukan dengan cara menentukan tujuan aktor dalam menggunakan sistem. Penentuan tujuan aktor dilakukan dengan melakukan analisis terhadap hasil identifikasi kebutuhan pengguna dan analisis hasil identifikasi fitur yang telah dilakukan pada tahap analisis persyaratan.

Spesifikasi untuk setiap *use case* yang teridentifikasi berisi penjelasan singkat, kondisi yang dibutuhkan sebelum *use case* dilakukan dan bagaimana kondisi setelah *use case* dilakukan, serta urutan tahap yang harus dilalui untuk menyelesaikan *use case*. Alur tahap *use case* yang terdapat pada spesifikasi *use case* akan menjadi informasi untuk pemodelan aktivitas ke dalam *activity diagram*. Penomoran *use case* dapat ditunjukan pada Tabel 4.11.

Tabel 4.11 Penomoran *use case*

Kode *Use Case*	Nama *Use Case*
UC-1	*Login*
UC-2	Mengelola pekerjaan
UC-3	Mengelola Pegawai
UC-4	Mengelola dokumentasi
UC-5	Mengelola kehadiran
UC-6	Mengelola keuangan
UC-7	Melihat riwayat sistem
UC-8	Mengelola pengguna

4.5.1 Use Case Diagram

Use case yang teridentifikasi kemudian dihubungkan dengan solusi yang diusulkan dalam aktivitas proses bisnis *to-be*, dengan pemangku kepentingan, dan dengan fitur yang telah teridentifikasi. Penghubungan *use case* dengan aktivitas proses bisnis *to-be* dilakukan untuk menunjukan ketelusuran yang lebih baik dalam penggunaan model proses bisnis dan diagram *use case* dalam proses pengembangan sistem informasi. Hubungan *use case* dengan aktivitas proses bisnis *to-be* dapat ditunjukan pada Tabel 4.12.

Tabel 4.12 Hubungan aktivitas proses bisnis *to-be* dengan *use case*

Proses Bisnis	Aktivitas	*Use Case*
Pembuatan laporan keuangan	Menghitung total pendapatan gaji	Mengelola keuangan
	Menyusun laporan keuangan	
Pembuatan dokumentasi pekerjaan	Mengkompilasi dokumentasi pekerjaan	Mengelola dokumentasi

Kemudian dilakukan penghubungan pemangku kepentingan dengan *use case* untuk membantu pemangku kepentingan memahami apa saja yang dapat dilakukan pengguna melalui sistem informasi yang akan dikembangkan. Hubungan *use case* dengan pemangku kepentingan terdapat pada Tabel 4.13.

Tabel 4.13 Hubungan *use case* dengan pemangku kepentingan

Use Case	**Pengguna**	**Tipe Pemangku Kepentingan**
Login	Petugas Admin, Direktur	Pengguna
Mengelola pekerjaan		
Mengelola pegawai		
Mengelola dokumentasi		
Mengelola kehadiran		
Mengelola keuangan		
Melihat riwayat sistem		
Mengelola pengguna	Direktur, Peneliti	Pengguna, Pengembang

Setiap *use case* kemudian dihubungkan dengan fitur-fitur yang teridentifikasi. Penghubungan ini dilakukan untuk mempertegas bahwa hasil pemodelan *use case* sesuai dengan hasil analisis persyaratan. Hubungan *use case* dengan fitur ditunjukan pada Tabel 4.14.

Tabel 4.14 Hubungan *use case* dengan fitur

Use Case	**Fitur**
Login	F-S-1
Mengelola pekerjaan	F-S-2
Mengelola pegawai	F-S-3
Mengelola dokumentasi	F-S-4
Mengelola kehadiran	F-S-5
Mengelola keuangan	F-S-6
Melihat riwayat sistem	F-S-7
Mengelola pengguna	F-S-8

4.5.2 Deskripsi Aktor

Deskripsi aktor merupakan penjelasan singkat mengenai tanggung jawab aktor yang telah teridentifiaksi. Di dalam identifikasi aktor terdapat informasi mengenai

tujuan aktor dalam menggunakan sistem informasi yang akan dikembangkan. Deskripsi aktor dapat ditunjukan pada Tabel 4.15.

Tabel 4.15 Deskripsi aktor

Nama Aktor	Deskripsi
Petugas Admin	Petugas perusahaan yang bertugas untuk mengelola pekerjaan, mengelola pegawai, mengelola dokumentasi, mengelola kehadiran, dan mengelola keuangan. Aktor petugas admin memilki akses terbatas terhadap sistem.
Direktur	Pengawas jalanya proses administrasi pekerjaan dan memiliki kedudukan tertinggi pada struktur organisasi perusahaan. Aktor direktur memilki semua akses terhadap sistem.

4.6 Spesifikasi *Use Case*

Pada bagian ini setiap *use case* akan dijelaskan secara lebih dalam. Tabel spesifikasi *use case* terdiri dari kode *use case* (*use case id*), kode fitur (*feature id*), deskripsi singkat (*brief description*), aktor yang menggunakan (*actor*), alur utama (*Basic flows of event*), alur alternatif (*alternatif flows*) yang dilakukan ketika menggunakan *use case* tersebut, *subflow*,kondisi sebelum *use case* berjalan (*pre-conditions*), dan kondisi setelah *use case* diajalankan (*post-conditions*). Tahap-tahap spesifikasi *use case* akan digunakan sebagai panduan memodelkan aktivitas ke dalam *activity diagram.*

4.6.1 Spesifikasi *Use Case* Login

Tabel 4.16 Spesifikasi *use case login*

Use Case Id	UC-1
Feature Id	F-S-1
Brief Description	*Use case login* mendeskripsikan bagaimana aktor melakakukan verivikasi terhadap identitasnya, sehingga memperoleh hak akses sistem yang sesuai.
Actor	Petugas Admin atau Direktur
Basic Flows of Events	**{*use case* dimulai}** 1. *Use case* dimulai ketika aktor memasukan *username* dan *password.* **{mengisi formulir identitas}** 2. Aktor mengirim *username* dan *password* kepada sistem. **{membatalkan pengiriman}** 3. Sistem melakukan identifikasi pengguna. **{mengidentifikasi pengguna}**

	4. Sistem menyimpan informasi identitas aktor. 5. Sistem menampilkan informasi berdasarkan hak akses aktor. **{*Use case* selesai}**
Alternatif Flows	**A1. Menangani kegagalan identifikasi pengguna** Pada **{mengindentifikasi pengguna}** di *basic flow* jika sistem mengalami kegagalan identifikasi pengguna, maka sistem menampilkan pesan bahwa identifikasi pengguna gagal, kemudian *use case* selesai. **A2. Menangani pengiriman formulir** Pada **{membatalkan pengiriman}** di *basic flow* jika aktor tidak mengirim formulir, maka *use case* dinyatakan selesai. **A5. Menangani kesalahan pengisian formulir** Pada **{mengisi formulir identitas}** di *basicflow* jika aktor tidak lengkap atau tidak benar pada saat mengisi formulir identitas, maka sistem menampilkan pesan untuk mengisi formulir yang belum terisi dengan benar.
Pre-Conditions	• Perangkat yang digunakan untuk mengakses sistem terhubung dengan internet. • Sistem terhubung dengan *server*.
Post-Conditions	• Identitas aktor teridentifikasi oleh sistem. • Aktor berhasil masuk ke dalam sistem (*logged in*). • Sistem menampilkan informasi sesuai hak akses aktor.

4.6.2 Spesifikasi *Use Case* Mengelola Pekerjaan

Tabel 4.17 Spesifikasi *use case* mengelola pekerjaan

Use Case Id	UC-2
Feature Id	F-S-2
Brief Description	*Use case* ini mendeskripsikan bagaimana aktor menggunakan sistem untuk menambah pekerjaan, menghapus pekerjaan, dan mengganti status pekerjaan.
Actor	Petugas Admin atau Direktur
Basic Flows of Events	**{*use case* dimulai}** 1. *Use case* dimulai ketika sistem menampilkan data pekerjaan. **{menampilkan data pekerjaan}**

	2. Aktor memilih untuk menambahkan data pekerjaan. 3. Sistem menampilkan formulir menambah data pekerjaan. 4. Lakukan *subflow* **Menambah pekerjaan.** **{*Use case* selesai}**
Alternatif Flows	**A1. Menghapus data pekerjaan** Pada **{menampilkan data pekerjaan}** di *basic flow* jika aktor memilih untuk menghapus data pekerjaan, maka lakukan *subflow* **Menghapus data pekerjaan**. **A2. Mengubah status pekerjaan** Pada **{menampilkan data pekerjaan}** di *basic flow* jika aktor memilih untuk mengubah status pekerjaan, maka lakukan *subflow* **Mengubah data pekerjaan**. **A3. Menangani kegagalan penghapusan data** Pada **{memproses penghapusan pekerjaan}** di *sub flow* jika sistem tidak menemukan data yang dimaksud, maka sistem menampilkan pesan data berhasil dihapus, kemudian *use case* selesai. **A4. Menangani kegagalan perubahan data** Pada **{memproses perubahan data pekerjaan}** di *subflow* jika sistem tidak menemukan data yang dimaksud, maka sistem menampilkan pesan status pekerjaan berhasil dirubah, kemudian *use case* selesai. **A5. Menangani kesalahan pengisian formulir** Pada **{mengisi formulir data pekerjaan}** di *subflow* jika aktor tidak lengkap atau tidak benar pada saat mengisi formulir, maka sistem menampilkan pesan untuk mengisi formulir yang belum terisi dengan benar. **A6. Menangani formulir tidak dikirimkan kepada sistem** Pada **{membatalkan pengiriman data pekerjaan}** di *subflow* jika aktor tidak mengirimkan atau keluar dari tampilan formulir maka *use case* dinyatakan selesai.
Subflow	***S1.* Menambah pekerjaan** 1. Aktor memasukan data pekerjaan. **{mengisi formulir data pekerjaan}** 2. Aktor mengirim data pekerjaan kepada sistem. **{membatalkan pengiriman data pekerjaan}**

	3. Sistem melakukan proses penyimpanan data pekerjaan. 4. Sistem menampilkan pesan penyimpanan data berhasil. 5. *Use case* selesai. **S2. Mengubah data pekerjaan** 1. Sistem menampilkan formulir perubahan pekerjaan 2. Aktor memasukan data pekerjaan. **{mengisi formulir data pekerjaan}** 3. Aktor mengirim data pekerjaan baru kepada sistem. **{membatalkan pengiriman data pekerjaan}** 4. Sistem melakukan proses penyimpanan perubahan data pekerjaan. **{memproses perubahan data pekerjaan}** 5. Sistem menampilkan pesan bahwa status pekerjaan berhasil dirubah. 6. *Use case* selesai. **S3. Menghapus data pekerjaan** 1. Sistem menampilkan formulir penghapusan data pekerjaan. 2. Aktor memasukan data pekerjan. **{mengisi formulir data pekerjaan}** 3. Aktor mengirim data pekerjaan ke dalam sistem. **{membatalkan pengiriman data pekerjaan}** 4. Sistem melakukan proses penghapusan data pekerjaan. **{memproses penghapusan pekerjaan}** 5. Sistem menampilkan pesan bahwa data pekerjaan berhasil di hapus. 6. *Use case* selesai.
Pre-Conditions	• Perangkat yang digunakan untuk mengakses sistem terhubung dengan internet. • Sistem terhubung dengan *server*. • Aktor berhasil masuk ke dalam sistem (*logged in*).
Post-Conditions	• Sistem berhasil menyimpan data hasil pengelolaan pekerjaan.

	• Aktor dapat melihat hasil pengelolaan data pekerjaan.

4.6.3 Spesifikasi *Use Case* Mengelola Pegawai

Tabel 4.18 Spesifikasi *use case* mengelola pegawai

Use Case Id	UC-3
Feature Id	F-S-3
Brief Description	*Use case* ini mendeskripsikan bagaimana aktor menggunakan sistem untuk menambah data pegawai, mengubah data pegawai, menghapus data pegawai, mencetak data pegawai, dan mengalokasikan pegawai ke dalam pekerjaan.
Actor	Petugas Admin atau Direktur
Basic Flows of Events	**{*use case* dimulai}** 1. *Use case* dimulai ketika sistem menampilkan data pegawai. **{menampilkan data pegawai}** 2. Aktor memilih untuk mengalokasikan data pegawai ke dalam pekerjaan. 3. Sistem menampilkan data alokasi pegawai. 4. Lakukan *subflow* **Alokasi pegawai.** **{*use case* selesai}**
Alternatif Flows	**A1. Manambah data pegawai** Pada **{menampilkan data pegawai}** di *basic flow* jika aktor memilih untuk menambah data pegawai, maka lakukan *subflow* **Menambah data pegawai.** **A2. Mengubah data pegawai** Pada **{menampilkan data pegawai}** di *basic flow* jika aktor memilih untuk mengubah data pegawai, maka lakukan *subflow* **Mengubah data pegawai.** **A3. Menghapus data pegawai** Pada **{menampilkan data pegawai}** di *basic flow* jika aktor memilih untuk menghapus data pegawai, maka lakukan *subflow* **Menghapus data pegawai.** **A4. Mencetak data pegawai** Pada **{menampilkan data pegawai}** di *basic flow* jika aktor memilih untuk mencetak data pegawai, maka lakukan *subflow* **Mencetak data pegawai.** **A5. Menangani kondisi pemilihan pegawai**

	Pada **{memilih pegawai}** di *subflow* jika aktor tidak memilih pegawai, maka sistem menampilkan pesan tidak ada pegawai dipilih. **A6. Menangani pengiriman formulir** Pada **{mengirim data pegawai dan pekerjaan}, {membatalkan pengiriman data pekerjaan}, {aktor tidak mengkonfirmasi penghapusan}** di *subflow* jika aktor tidak mengirimkan atau keluar dari formulir, maka *use case* dinyatakan selesai. **A7. Menangani kesalahan pengisian formulir** Pada **{mengisi formulir data pegawai}** di *subflow* jika aktor tidak lengkap atau tidak benar pada saat mengsi formulir, maka sistem menampilkan pesan untuk mengisi formulir yang belum terisi dengan benar.
Subflow	**S1. Alokasi pegawai** 1. Aktor memilih pegawai untuk di alokasi ke dalam pekerjaan. 2. Aktor memilih pekerjaan. 3. Aktor mengirim data pegawai dan pekerjaan ke dalam sistem. **{mengirim data pegawai dan pekerjaan}** 4. Sistem melakukan proses alokasi pegawai. 5. Sistem menampilkan pesan alokasi berhasil di berikan. **{memilih pegawai}** 6. *Use case* selesai. **S2. Menambah data pegawai** 1. Sistem menampilkan formulir menambah data pegawai. 2. Aktor memasukan data pegawai. **{mengisi formulir data pegawai}** 3. Aktor mengirim data pegawai ke dalam sistem. **{membatalkan pengiriman data pekerjaan}** 4. Sistem melakukan proses penyimpanan data pegawai. 5. Sistem menampilkan pesan pegawai berhasil ditambahkan. 6. *Use case* selesai. **S3. Mengubah data pegawai**

	1. Sistem menampilkan data pegawai yang akan dirubah. 2. Aktor mengubah data pegawai. **{mengisi formulir data pegawai}** 3. Aktor mengirim hasil perubahan ke dalam sistem. **{membatalkan pengiriman data pekerjaan}** 4. Sistem melakukan proses perubahan data. 5. Sistem menampilkan pesan bahwa data pegawai berhasil dirubah. 6. *Use case* selesai. **S4. Menghapus data pegawai** 1. Sistem menampilkan pesan konfirmasi penghapusan data pegawai. 2. Aktor melakukan konfirmasi penghapusan data pegawai. **{aktor tidak mengkonfirmasi penghapusan}** 3. Sistem memproses penghapusan data pegawai. 4. Sistem menampilkan pesan bahwa data pegawai berhasil dihapus. 5. *Use case* selesai. **S5. Mencetak data pegawai** 1. Sistem melakukan proses pencetakan data pegawai. 2. *Use case* selesai.
Pre-Conditions	• Perangkat yang digunakan untuk mengakses sistem terhubung dengan internet. • Sistem terhubung dengan *server*. • Aktor berhasil masuk ke dalam sistem (*logged in*). • Data pekerjaan telah tersimpan di dalam sistem
Post-Conditions	• Sistem berhasil menyimpan data hasil pengelolaan pegawai. • Aktor dapat melihat hasil pengelolaan data pegawai.

4.6.4 Spesifikasi *Use Case* Mengelola Dokumentasi

Tabel 4.19 Spesifikasi *use case* mengelola dokumentasi

Use Case Id	UC-4
Feature Id	F-S-4

Brief Description	*Use case* ini mendeskripsikan bagaimana aktor menggunakan sistem untuk menambah data dokumentasi, mengubah data dokumentasi, menghapus data dokumentasi, dan mencetak data dokumentasi.
Actor	Petugas Admin atau Direktur
Basic Flows of Events	**{*Use case* dimulai}** 1. *Use case* dimulai ketika sistem menampilkan data dokumentasi. **{menampilkan data dokumentasi}** 2. Aktor memilih untuk menambahkan data dokumentasi pekerjaan. 3. Sistem menampilkan formulir penambahan data dokumentasi pekerjaan. 4. Lakukan *subflow* **Menambah data dokumentasi**. **{*Use case* selesai}**
Alternatif Flows	**A1. Mengubah data dokumentasi** Pada **{menampilkan data dokumentasi}** di *basic flow* jika aktor memilih untuk mengubah data dokumentasi, maka lakukan *subflow* **Mengubah data dokumentasi.** **A2. Menghapus data dokumentasi** Pada **{menampilkan data dokumentasi}** di *basic flow* jika aktor memilih untuk menghapus data dokumentasi, maka lakukan *subflow* **Menghapus data dokumentasi.** **A3. Mencetak data dokumentasi** Pada **{menampilkan data dokumentasi}** di *basic flow* jika aktor memilih untuk mencetak data dokumentasi, maka lakukan *subflow* **Mencetak data dokumentasi.** **A4. Menangani pembatalan pengiriman formulir** Pada **{membatalkan pengiriman data dokumentasi}, {membatalkan konfirmasi penghapusan}** di *subflow* jika aktor tidak mengirimkan atau keluar dari formulir, maka *use case* dinyatakan selesai. **A5. Menangani kesalahan pengisian formulir** Pada **{mengisi formulir data dokumentasi}** di *subflow* jika aktor tidak lengkap atau tidak benar pada saat mengisi formulir, maka sistem menampilkan pesan untuk mengisi formulir yang belum terisi dengan benar.

Subflow	**S1. Menambah data dokumentasi** 1. Aktor memasukan data dokumentasi pekerjaan. **{mengisi formulir data dokumentasi}** 2. Aktor mengirim data dokumentasi pekerjaan ke dalam sistem. **{membatalkan pengiriman data dokumentasi}** 3. Sistem melakukan proses penyimpanan data dokumentasi pekerjaan. 4. Sistem menampilkan pesan dokumentasi berhasil ditambahkan. 5. *Use case* selesai. **S2. Mengubah data dokumentasi** 1. Sistem menampilkan data dokumentasi yang akan dirubah. 2. Aktor mengubah data dokumentasi pekerjaan. **{mengisi formulir data dokumentasi}** 3. Aktor mengirim data dokumentasi pekerjaan baru ke dalam sistem. **{membatalkan pengiriman data dokumentasi}** 4. Sistem melakukan proses penyimpanan perubahan data. 5. Sistem menampilkan pesan bahwa perubahan dokumentasi berhasil disimpan. 6. *Use case* selesai. **S3. Menghapus data dokumentasi** 1. Sistem menampilkan pesan konfirmasi penghapusan. 2. Aktor melakukan konfirmasi penghapusan data dokumentasi pekerjaan. **{membatalkan konfirmasi penghapusan}** 3. Sistem melakukan proses penghapusan data dokumentasi 4. Sistem menampilkan pesan bahwa data dokumentasi berhasil dihapus. 5. *Use case* selesai. **S4. Mencetak data dokumentasi** 1. Sistem melakukan proses pencetakan data dokumentasi pekerjaan.

	2. *Use case* selesai.
Pre-Conditions	• Perangkat yang digunakan untuk mengakses sistem terhubung dengan internet. • Sistem terhubung dengan *server*. • Aktor berhasil masuk ke dalam sistem (*logged in*). • Data pekerjaan telah tersimpan di dalam sistem.
Post-Conditions	• Sistem berhasil menyimpan data hasil pengelolaan dokumentasi pekerjaan. • Aktor dapat melihat hasil pengelolaan data dokumentasi pekerjaan.

4.6.5 Spesifikasi *Use Case* Mengelola Kehadiran

Tabel 4.20 Spesifikasi *use case* mengelola kehadiran

Use Case Id	UC-5
Feature Id	F-S-5
Brief Description	*Use case* mendeskripsikan bagaimana aktor menggunakan sistem untuk menambah data kehadiran, membatalkan data kehadiran, mencetak rekap kehadiran.
Actor	Petugas Admin atau Direktur
Basic Flows of Events	**{*use case* dimulai}** 1. Use case dimulai ketika aktor memilih untuk menambah data kehadiran pegawai. 2. Sistem menampilkan daftar pekerjaan. **{menampilkan daftar pekerjaan}** 3. Aktor memilih pekerjaan. 4. Sistem menampilkan data pegawai pada satu pekerjaan. **{menampilkan data pegawai}** 5. Lakukan *subflow* **Menambah data kehadiran**. **{*use case* selesai}**
Alternatif Flows	**A1. Menangani pengambilan daftar pekerjaan** Pada **{menampilkan daftar pekerjaan}** di *basic flow* jika aktor belum menambahkan data pekerjaan, maka daftar pekerjaan akan kosong dan *use case* dinyatakan selesai. **A2. Menangani pengambilan data pegawai** Pada **{menampilkan data pegawai}** di *basic flow* jika aktor belum mengalokasikan pegawai pada pekerjaan

	tersebut, maka daftar pegawai akan kosong dan *use case* dinyatakan selesai. **A3. Menangani pemilihan pegawai** Pada **{memilih pegawai untuk di kirim}** di *subflow* jika aktor tidak memilih satupun pegawai, maka tidak ada data yang diproses dan melanjutkan ke langkah berikutnya. **A4. Membatalkan data kehadiran pegawai** Pada **{menampilkan data pegawai}** di *basic flow* jika aktor memilih untuk membatalkan data kehadiran pegawai, maka lakukan *subflow* **Membatalkan data kehadiran**. **A5. Mencetak data kehadiran pegawai** Pada **{menampilkan data pegawai}** di *basic flow* jika aktor memilih untuk mencetak data kehadiran pegawai, maka lakukan *subflow* **Mencetak data kehadiran.**
Subflow	**S1. Menambah data kehadiran** 1. Aktor memilih pegawai yang akan di tambahkan ke daftar kehadiran. **{memilih pegawai untuk di kirim}** 2. Aktor mengirim daftar pegawai ke dalam sistem. 3. Sistem memproses penambahan data kehadiran pegawai. 4. Sistem menampilkan pesan bahwa kehadiran berhasil ditambahkan. 5. Sistem menampilkan data rekap kehadiran pegawai 6. *Use case* selesai. **S2. Membatalkan data kehadiran** 1. Aktor memilih untuk melihat rekap kehadiran pegawai. 2. Sistem menampilkan detail rekap kehadiran pegawai 3. Aktor memilih pegawai yang akan di batalkan kehadiranya. **{memilih pegawai untuk dikirim}** 4. Aktor mengirim daftar pegawai ke dalam sistem. 5. Sistem memproses pembatalan kehadiran pegawai. 6. Sistem menampilkan pesan bahwa kehadiran berhasil di hapus.

	7. *Use case* selesai. **S3. Mencetak data kehadiran** 1. Aktor memilih untuk melihat rekap kehadiran pegawai. 2. Aktor meminta sistem untuk mencetak rekap kehadiran pegawai. 3. Sistem memproses pencetakan rekap kehadiran pegawai. 4. *Use case* selesai.
Pre-Conditions	• Perangkat yang digunakan untuk mengakses sistem terhubung dengan internet. • Sistem terhubung dengan *server*. • Aktor berhasil masuk ke dalam sistem (*logged in*). • Data alokasi pegawai telah tersimpan di dalam sistem.
Post-Conditions	• Sistem berhasil menyimpan data hasil pengelolaan kehadiran pegawai. • Aktor dapat melihat hasil pengelolaan data kehadiran pegawai.

4.6.6 Spesifikasi *Use Case* Mengelola Keuangan

Tabel 4.21 Spesifikasi *use case* mengelola keuangan

Use Case Id	UC-6
Feature Id	S-F-6
Brief Description	*Use case* ini mendeskripsikan bagaimana aktor menggunakan sistem untuk menambah data keuangan, membatalkan data keuangan, mencetak data keuangan, dan mengatur rumus keuangan.
Actor	Petugas Admin atau Direktur
Basic Flows of Events	**{*use case* dimulai}** 1. Use case dimulai ketika aktor memilih untuk melihat data keuangan pegawai. 2. Sistem menampilkan daftar pekerjaan. **{menampilkan daftar pekerjaan}** 3. Aktor memilih pekerjaan. 4. Sistem menampilkan data keuangan seluruh pegawai pada pekerjaan. **{menampilkan data keuangan}**

	{*use case* selesai}
Alternatif Flows	**A1. Menangani pengambilan daftar pekerjaan** Pada **{menampilkan daftar pekerjaan}** di *basic flow* jika aktor belum menambahkan data pekerjaan, maka daftar pekerjaan akan kosong dan *use case* dinyatakan selesai. **A2. Menambah data keuangan** Pada **{menampilkan data keuangan}** di *basic flow* jika aktor memilih untuk menambahkan data keuangan, maka lakukan *sub flow* **Menambah data keuangan.** **A3. Menangani pemilih jenis** Pada **{memilih jenis rincian keuangan}** di *subflow* aktor bisa memilih antara tambahan keuangan atau pengurangan keuangan. **A4. Membatalkan data keuangan** Pada **{menampilkan data keuangan}** di *basic flow* jika aktor memilih untuk membatalkan data keuangan, maka lakukan *sub flow* **Membatalkan data keuangan.** **A5. Menangani pembatalan pengiriman formulir** Pada **{membatalkan pengiriman data formulir}, {membatalkan konfirmasi penghapusan}** di *subflow* jika aktor tidak mengirimkan atau keluar dari formulir, maka *use case* dinyatakan selesai. **A6. Mencetak data keuangan** Pada **{menampilkan data keuangan}** di *basic flow* jika aktor memilih untuk mencetak data keuangan, maka lakukan *sub flow* **Mencetak data keuangan.** **A7. Mengatur rumus keuangan** Pada **{menampilkan data keuangan}** di *basic flow* jika aktor memilih untuk mengatur rumus keuangan, maka lakukan *sub flow* **Mengatur rumus keuangan.** **A8. Menangani kesalahan pengisian formulir** Pada **{mengisi formulir penambahan keuangan}, {mengisi formulir rumus keuangan}** di *subflow* jika aktor tidak lengkap atau tidak benar pada saat mengisi formulir, maka sistem menampilkan pesan untuk mengisi formulir yang belum terisi dengan benar benar.
Subflow	**S1. Menambah data keuangan** 1. Aktor memilih rincian data keuangan dari pegawai. **{memilih jenis rincian keuangan}**

	2. Sistem menampilkan rincian keuangan. 3. Aktor memilih untuk menambah data keuangan. 4. Sistem menampilkan formulir penambahan keuangan. 5. Aktor memasukan data penambahan keuangan. **{mengisi formulir penambahan keuangan}** 6. Aktor mengirim data penambahan keuangan ke dalam sistem. **{membatalkan pengiriman data formulir}** 7. Sistem melakukan proses penambahan keuangan. 8. Sistem menampilkan pesan penambahan keuangan berhasil diberikan. 9. *Use case* selesai. **S2. Membatalkan data keuangan** 1. Aktor memilih rincian data keuangan dari pegawai. **{memilih jenis rincian keuangan}** 2. Sistem menampilkan rincian keuangan. 3. Aktor memilih untuk membatalkan data keuangan. 4. Sistem menampilkan pesan konfirmasi penghapusan. 5. Aktor mengkonfirmasi penghapusan data keuangan. **{membatalkan konfirmasi penghapusan}** 6. Sistem melakukan proses penghapusan data keuangan. 7. Sistem menampilkan pesan bahwa data keuangan berhasil di hapus. 8. *Use case* selesai. **S3. Mencetak data keuangan** 1. Sistem melakukan proses pencetakan data keuangan. 2. *Use case* selesai. **S4. Mengatur rumus keuangan** 1. Sistem menampilkan daftar rumus keuangan berdasarkan pekerjaan. **{menampilkan daftar pekerjaan}** 2. Aktor memilih untuk mengatur rumus keuangan. 3. Sistem menampilkan data rumus keuangan yang akan di rubah.

	4. Aktor mengubah data rumus keuangan. **{mengisi formulir rumus keuangan}** 5. Aktor mengirim data rumus keuangan baru ke dalam sistem. **{membatalkan pengiriman data formulir}** 6. Sistem melakukan proses perubahan data. 7. Sistem menampilkan pesan bahwa rumus berhasil dirubah. 8. *Use case* selesai.
Pre-Conditions	• Perangkat yang digunakan untuk mengakses sistem terhubung dengan internet. • Sistem terhubung dengan *server*. • Aktor berhasil masuk ke dalam sistem (*logged in*). • Data kehadiran pegawai telah tersimpan di dalam sistem.
Post-Conditions	• Sistem berhasil menyimpan data hasil pengelolaan keuangan. • Aktor dapat melihat hasil pengelolaan data keuangan.

4.6.7 Spesifikasi *Use Case* Melihat Riwayat Sistem

Tabel 4.22 Spesifikasi *use case* melihat riwayat sistem

Use Case Id	UC-7
Feature Id	F-S-7
Brief Description	*Use case* ini mendeskripsikan bagaimana aktor menggunakan sistem untuk melihat riwayat aktivitas penggunaan sistem.
Actor	Petugas Admin atau Direktur
Basic Flows of Events	**{*use case* dimulai}** 1. *Use case* dimulai ketika aktor memilih untuk melihat riwayat penggunaan sistem. 2. Sistem menampilkan riwayat penggunaan. **{*use case* selesai}**
Alternatif Flows	-
Pre-Conditions	• Perangkat yang digunakan untuk mengakses sistem terhubung dengan internet. • Sistem terhubung dengan *server*.

	• Aktor berhasil masuk ke dalam sistem (*logged in).*
Post-Conditions	• Aktor dapat melihat riwayat aktivitas penggunaan sistem.

4.6.8 Spesifikasi *Use Case* Mengelola Pengguna

Tabel 4.23 Spesifikasi *use case* mengelola pengguna

Use Case Id	UC-8
Feature Id	F-S-8
Brief Description	*Use case* ini mendeskripsikan bagaimana aktor direktur menggunakan sistem untuk menambah pengguna, menghapus pengguna, dan mengubah data pengguna.
Actor	Direktur
Basic Flows of Events	**{*use case* dimulai}** 1. *Use case* dimulai ketika sistem menampilkan daftar pengguna. **{menampilkan daftar pengguna}** 2. Aktor direktur memilih untuk menambahkan data pengguna. 3. Lakukan *subflow* **Menambah data pengguna.** **{*use case* berakhir}**
Alternatif Flows	**A1. Menangani pembatalan pengiriman formulir** Pada **{membatalkan pengiriman data formulir}** di *subflow,* dan **{membatalkan konfirmasi penghapusan}** di *subflow* jika aktor direktur tidak mengirimkan atau keluar dari formulir, maka *use case* dinyatakan selesai. **A2. Menangani kesalahan pengisian formulir** Pada **{memasukan data formulir}** di *subflow* jika aktor direktur tidak lengkap atau tidak benar pada saat mengisi formulir, maka sistem menampilkan pesan untuk mengisi formulir yang belum terisi dengan benar. **A3. Menghapus data pengguna** Pada **{menampilkan daftar pengguna}** di *basic flow* jika aktor direktur memilih untuk menghapus data pengguna, maka lakukan *subflow* **Menghapus data pengguna.** **A4. Mengubah data pengguna**

	Pada **{menampilkan daftar pengguna}** di *basic flow* jika aktor direktur memilih untuk mengubah data pengguna, maka lakukan *subflow* **Mengubah data pengguna.**
Subflow	**S1. Menghapus data pengguna** 1. Sistem menampilkan pesan konfirmasi penghapusan data pengguna. 2. Aktor direktur mengkonfirmasi penghapusan data pengguna. **{membatalkan konfirmasi penghapusan}** 3. Sistem melakukan proses penghapusan data pengguna. 4. Sistem menampilkan pesan bahwa data pengguna berhasil di hapus. 5. *Use case* selesai. **S2. Mengubah data pengguna** 1. Sistem menampilkan data pengguna yang akan dirubah. 2. Aktor direktur mengubah data pengguna. **{memasukan data formulir}** 3. Aktor mengirim data pengguna ke dalam sistem. **{membatalkan pengiriman data formulir}** 4. Sistem melakukan proses perubahan data pengguna. 5. Sistem menampilkan pesan bahwa perubahan data berhasil di lakukan. 6. *Use case* selesai. **S3. Menambah data pengguna** 1. Sistem menampilkan formulir penambahan data pengguna. 2. Aktor direktur memasukan data penambahan pengguna. **{memasukan data formulir}** 3. Aktor direktur mengirim data pengguna ke dalam sistem. **{membatalkan pengiriman data formulir}** 4. Sistem melakukan proses penyimpanan data pengguna.

	5. Sistem menampilkan pesan bahwa pengguna berhasil di tambahkan. 6. *Use case* selesai.
Pre-Conditions	• Perangkat yang digunakan untuk mengakses sistem terhubung dengan internet. • Sistem terhubung dengan *server*. • Aktor direktur berhasil masuk ke dalam sistem (*logged in*).
Post-Conditions	• Sistem berhasil menyimpan data hasil pengelolaan pengguna. • Aktor direktur dapat melihat hasil pengelolaan data pengguna.

4.7 Pemodelan Aktivitas

Pemodelan aktivitas dalam bentuk *activity diagram* dilakukan berdasarkan alur yang terdapat pada spesifikasi *use case*. Pemodelan aktivitas pada beberapa alur *use case* dilakukan untuk mendokumentasikan alur *use case* sehingga dapat lebih mudah dipahami oleh pemangku kepentingan. Beberapa aktivitas yang dimodelkan pada penelitian ini diantaranya adalah aktivitas mengelola pegawai, mengelola dokumentasi, dan mengelola keuangan.

4.7.1 *Activity Diagram* Mengelola Pegawai

Activity Diagram mengelola pegawai pada Gambar 4.13 menunjukan aktivitas yang dilakukan oleh aktor dan sistem pada saat sistem digunakan oleh aktor untuk mengelola data pegawai. *Activity diagram* dibuat berdasarkan alur utama *use case* mengelola pegawai pada Tabel 4.18. Selain alur utama aktivitas mengalokasi pegawai, pada *activity diagram* juga memuat sub aktivitas lain yang di lakukan oleh aktor pada saat menggunakan sistem. Setiap sub aktivitas dimodelkan berdasarkan alur alternatif pada *use case* mengelola pegawai.

Sub aktivitas yang dimodelkan di antaranya adalah :

1. Sub aktivitas menambah data pegawai yang memvisualisasikan alur alternatif A1 (Menambah Data Pegawai).
2. Sub aktivitas mengubah data pegawai yang memvisualisasikan alur alternatif A2 (Mengubah Data Pegawai).
3. Sub aktivitas menghapus data pegawai yang memvisualisasikan alur alternatif A3 (Menghapus Data Pegawai).
4. Sub aktivitas mencetak data pegawai yang memvisualisasikan alur alternatif A4 (Mencetak data pegawai).

4.7.2 *Activity Diagram* Mengelola Dokumentasi

Activity diagram mengelola dokumentasi pada Gambar 4.14 menunjukan aktivitas yang dilakukan oleh aktor dan sistem pada saat sistem digunakan oleh aktor untuk mengelola data dokumentasi pekerjaan. *Activity diagram* dibuat berdasarkan alur utama *use case* mengelola dokumentasi pada Tabel 4.19 . Selain alur utama aktivitas menambah dokumentasi, pada diagram aktvitias juga memuat sub aktivitas lain yang di lakukan oleh aktor pada saat menggunakan sistem. Masing-masing sub aktivitas yang dimodelkan berdasarkan alur alternatif pada *use case* mengelola dokumentasi.

Sub aktivitas yang dimodelkan di antaranya adalah :

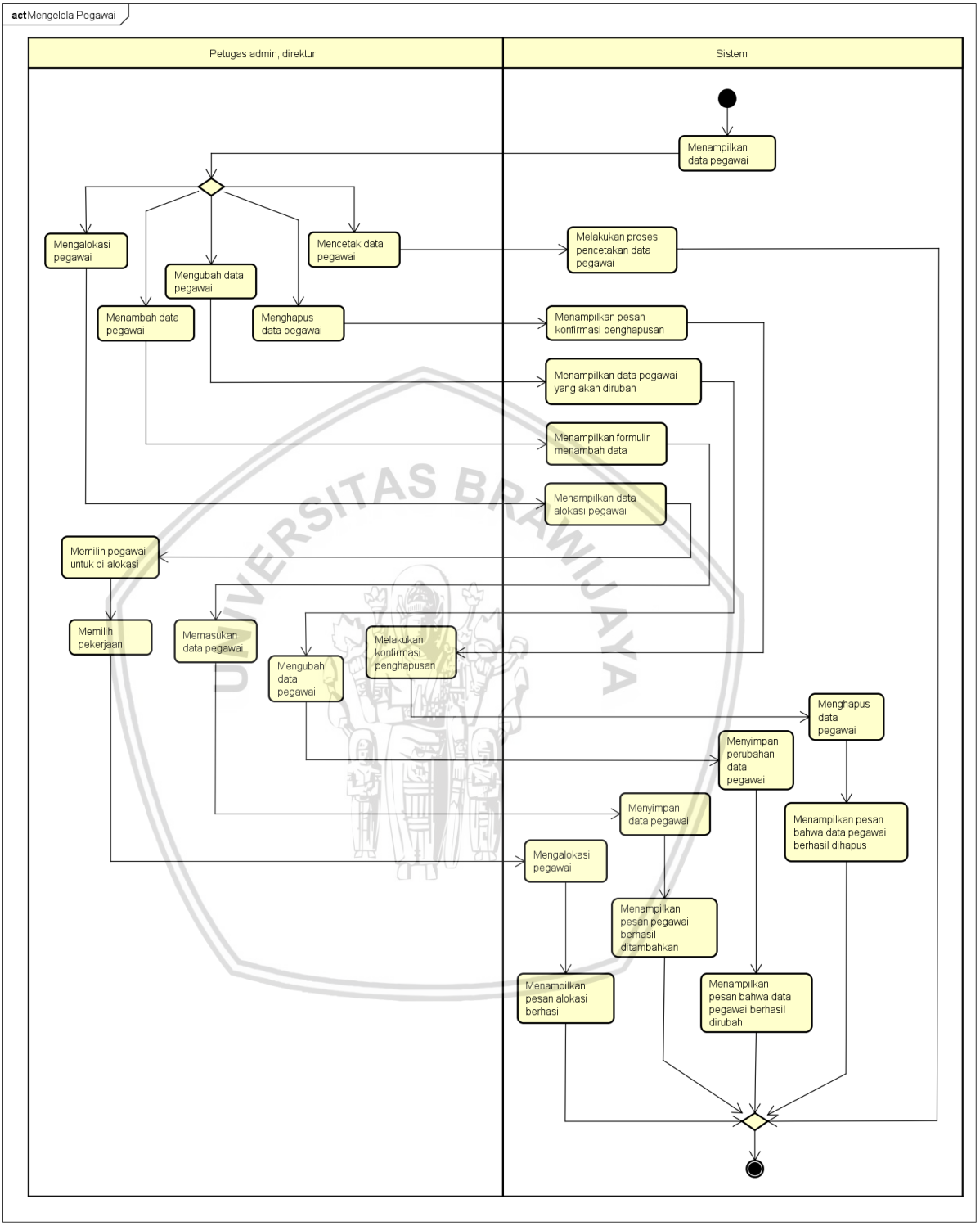
1. Sub aktivitas mengubah data dokumentasi yang memvisualisasikan alur alternatif A1 (Mengubah Data Dokumentasi).
2. Sub aktivitas menghapus data dokumentasi yang memvisualisasikan alur alternatif A2 (Menghapus Data Dokumentasi).
3. Sub aktivitas mencetak data dokumentasi yang memvisualisasikan alur alternatif A3 (Mencetak Data Dokumentasi).

4.7.3 *Activity Diagram* Mengelola Keuangan

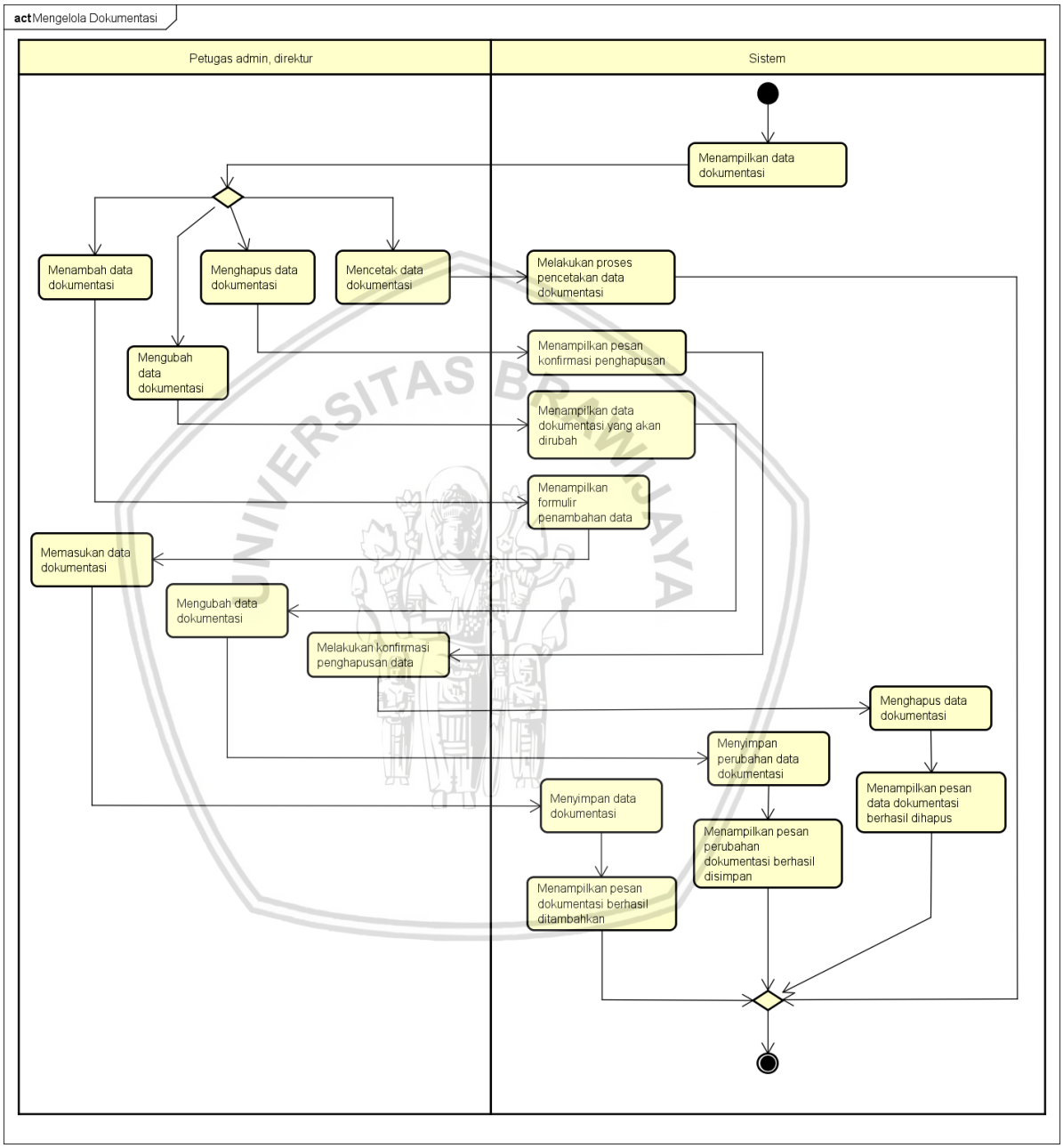
Activity diagram mengelola keuangan pada Gambar 4.15 menunjukan aktivitas yang dilakukan oleh aktor dan sistem pada saat sistem digunakan oleh aktor untuk mengelola data keuangan. *Activity diagram* dimodelkan berdasarkan alur utama *use case* mengelola keuangan pada Tabel 4.21. Selain alur utama aktivitas melihat data keuangan, pada *activity diagram* juga memuat sub aktivitas lain yang di lakukan oleh aktor pada saat menggunakan sistem. Masing - masing sub aktivitas dimodelkan berdasarkan alur alternatif pada *use case* mengelola keuangan.

Sub aktivitas yang dimodelkan berdasarkan alur alternatif *use case* mengelola keuangan di antaranya adalah :

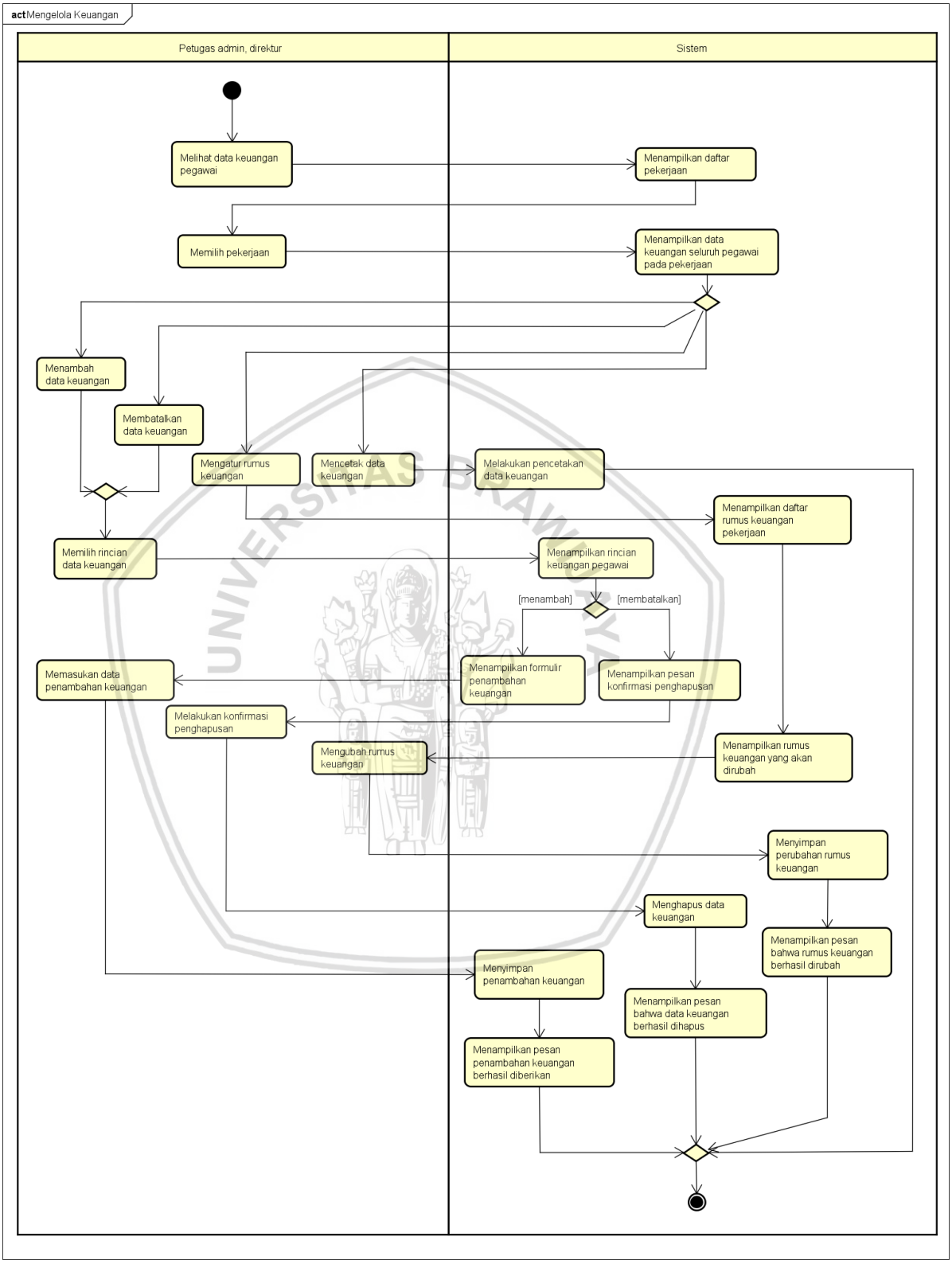
1. Sub aktivitas menambah data keuangan yang memvisualisasikan alur alternatif A2 (Menambah Data Keuangan).
2. Sub aktivitas membatalkan data keuangan yang memvisualisasikan alur alternatif A4 (Membatalkan Data Keuangan).
3. Sub aktivitas mencetak data keuangan yang memvisualisasikan alur alternatif A6 (Mencetak Data Keuangan).
4. Sub aktivitas mengatur rumus keuangan yang memvisualisasikan alur alternatif A7 (Mengatur Rumus Keuangan).



Gambar 4.13 *Activity diagram* mengelola pegawai



Gambar 4.14 *Activity diagram* mengelola dokumentasi



Gambar 4.15 *Activity diagram* mengelola keuangan

BAB 5 PERANCANGAN

Bab ini menjelaskan perancangan sistem yang dikembangkan sesuai hasil analisis persyaratan yang telah dilakukan sebelumnya. Hasil pemodelan yang dijelaskan pada bab ini di antaranya terdapat *sequence diagram, class diagram*, perancangan basis data yang didokumentasikan ke dalam bentuk *physical data model* (PDM), perancangan algoritme yang didokumentasikan ke dalam bentuk *pseducode*, dan perancangan antarmuka pengguna.

5.1 Pemodelan Interaksi Objek

Pemodelan interaksi objek dilakukan untuk memvisualisasikan pertukaran pesan antara entitas aktor, entitas *boundary*, objek *control,* dan objek *entity* yang saling berinteraksi untuk memenuhi kebutuhan pengguna sistem. *Sequence diagram* digunakan sebagai sarana dokumentasi pemodelan interaksi objek. *Sequence diagram* dibuat berdasarkan alur dari beberapa *use case* yang sudah didokumentasikan sebelumnya pada spesifikasi *use case.*

Beberapa *sequence diagram* yang dijelaskan pada bab ini merupakan visualisasi dari interaksi pada proses mengalokasi pegawai, menambah data dokumentasi pekerjaan, dan menampilkan data keuangan yang mengacu pada alur utama *use case* mengelola pegawai, mengelola dokumentasi, dan mengelola keuangan.

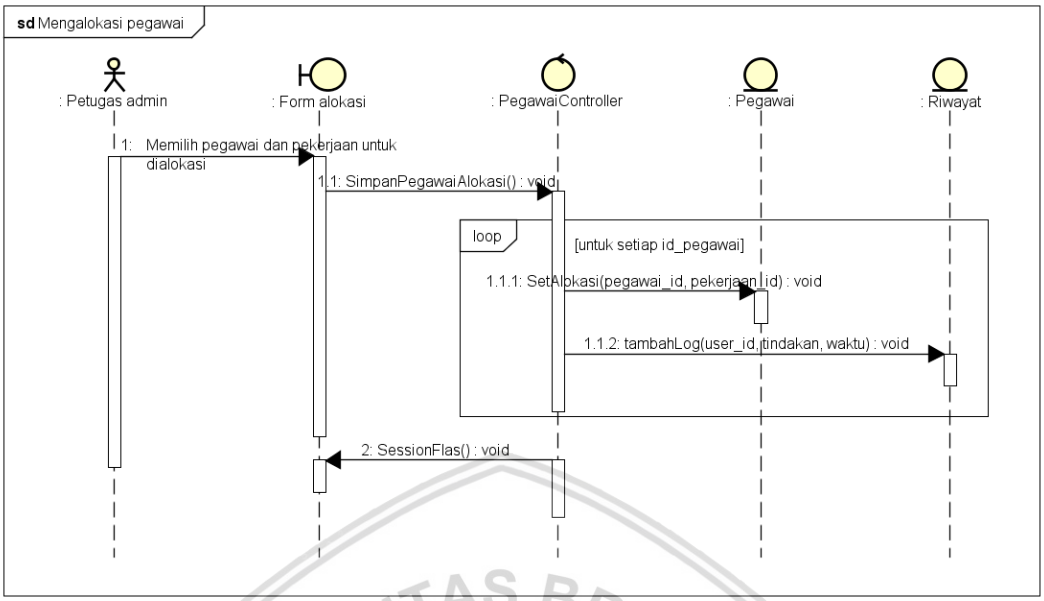
5.1.1 Mengalokasi Pegawai

Interaksi objek pada proses untuk mengalokasi pegawai ke dalam pekerjaan divisualisasikan ke dalam *sequence diagram* yang dapat dilihat dalam Gambar 5.1. Interaksi tersebut terjadi pada saat aktor petugas admin atau direktur menggunakan sistem untuk mengalokasikan data pegawai ke dalam pekerjaan dimana kedua data tersebut telah tersimpan di dalam sistem.

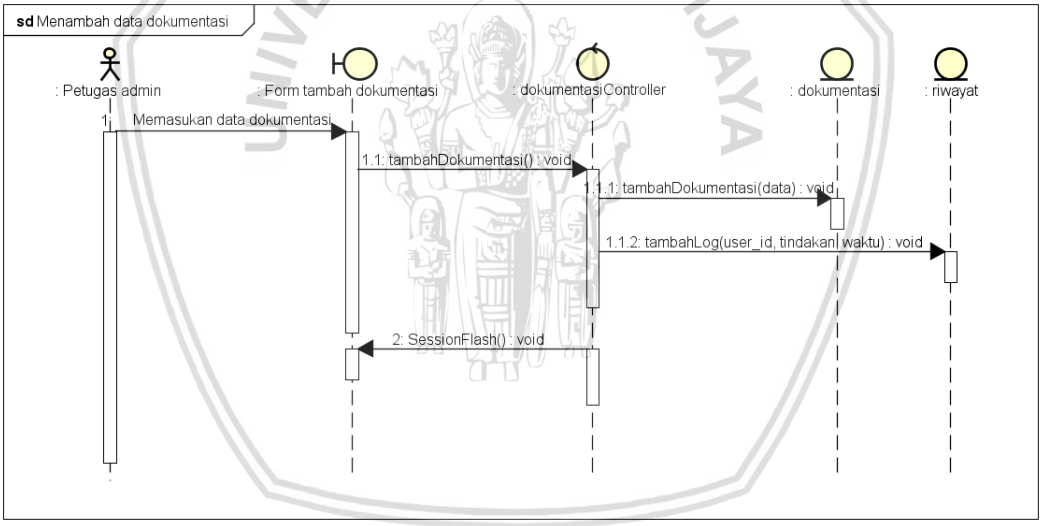
Beberapa objek yang terlibat dalam interaksi pada proses untuk mengalokasi pegawai ke dalam pekerjaan di antaranya adalah aktor petugas admin yang juga bisa mewakili direktur, *form* alokasi sebagai objek *boundary,* pegawaiController sebagai objek *control*, pegawai dan riwayat sebagai objek *entity*.

5.1.2 Menambah Data Dokumentasi Pekerjaan

Sequence diagram menambah data dokumentasi pekerjaan terdapat pada Gambar 5.2 memvisualisasikan interaksi antar objek yang terjadi pada saat aktor petugas admin atau direktur menggunakan sistem untuk menambahkan dokumentasi pekerjaan yang telah terselesaikan oleh petugas lapangan. Objek yang terlibat dalam interaksi pada proses menambah data dokumentasi pekerjaan di antaranya adalah aktor petugas admin yang juga mewakili aktor direktur, *form* tambah dokumentasi sebagai objek *boundary*, dokumentasiController sebagai objek *control*, dokumentasi dan riwayat sebagai objek *entity*.



Gambar 5.1 *Sequence diagram* mengalokasi pegawai



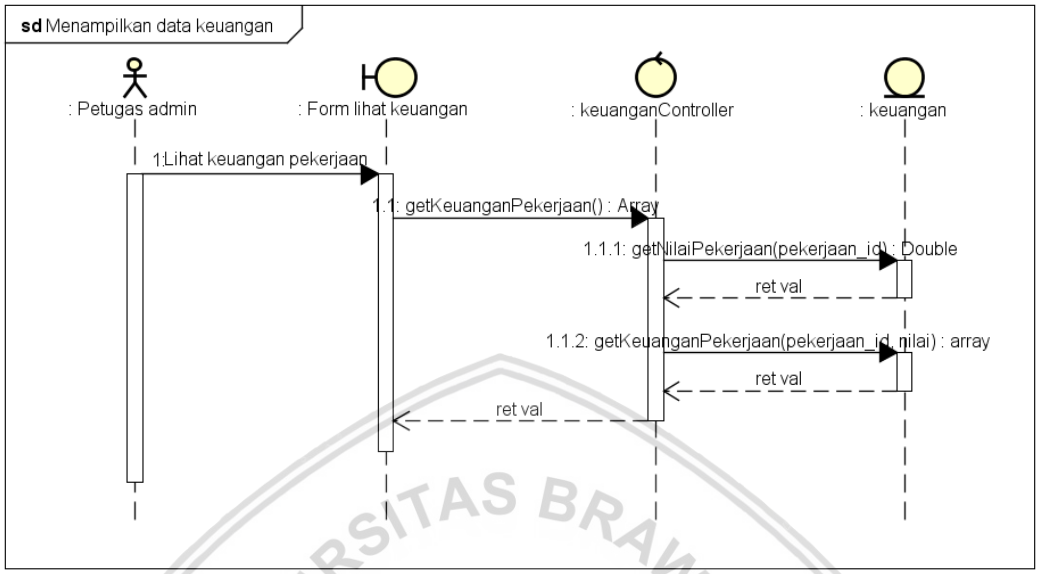
Gambar 5.2 *Sequence diagram* menambah data dokumentasi pekerjaan

5.1.3 Menampilkan Data Keuangan

Interaksi objek pada proses untuk menampilkan data keuangan divusalisasikan ke dalam *sequence diagram* yang dapat dilihat dalam Gambar 5.3. Interaksi tersebut terjadi pada saat aktor petugas admin atau aktor direktur menggunakan sistem untuk menampikan hasil perhitungan data keuangan pegawai dari suatu pekerjaan tertentu yang telah tersimpan dan dihitung secara otomatis di dalam sistem.

Beberapa objek yang terlibat dalam interaksi proses menampilkan data keuangan di antaranya adalah aktor petugas admin yang juga dapat mewakili aktor

direktur, *form* lihat keuangan sebagai objek *boundary*, keuanganController sebagai objek *control,* dan keuangan sebagai objek *entity*.



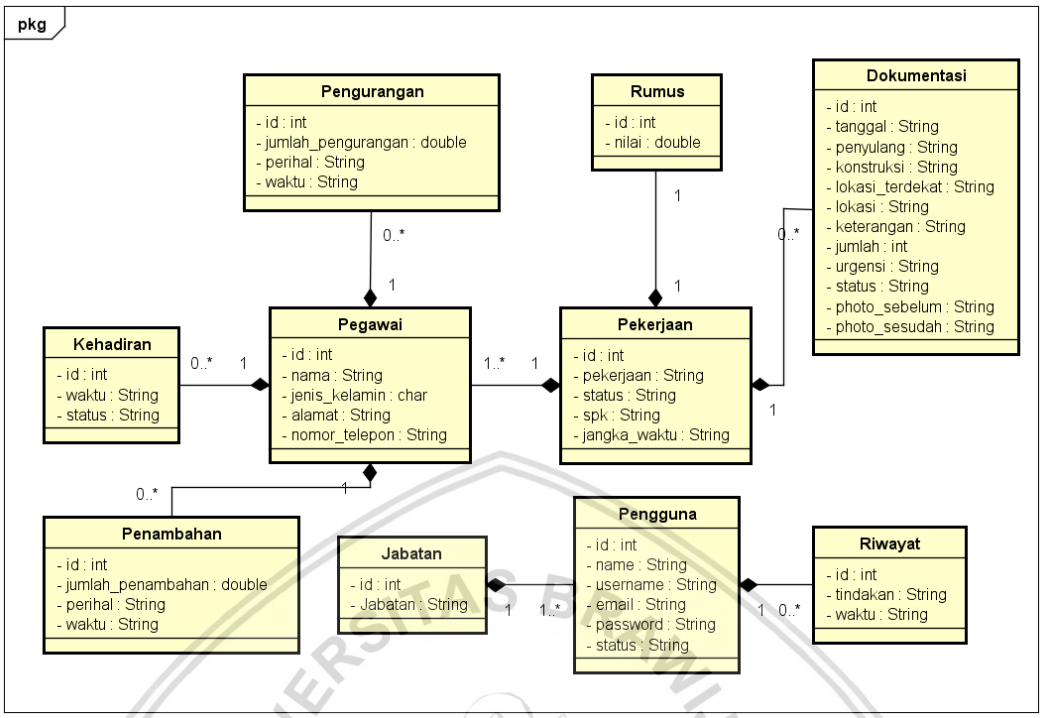
Gambar 5.3 *Sequence diagram* menampilkan data keuangan

5.2 Pemodelan Kelas

Pemodelan kelas dilakukan untuk mendokumentasikan sekelompok objek dari sistem yang akan dikembangkan. Pemodelan kelas divisualisasikan ke dalam bentuk *class diagram*. Pada bagian ini dijelaskan *class diagram* secara umum sebagai acuan dasar pengembangan sistem secara umum, sebagai acuan dalam perancangan basis data, dan sebagai acuan pemodelan kelas untuk kerangka kerja pemrogaman PHP. Pada pemodelan kelas kerangka kerja pemrogaman dijelaskan *class diagram* untuk kerangka kerja pemrogaman Laravel sebagai panduan implementasi pemrogaman sistem menggunakan kerangka kerja Laravel.

5.2.1 *Class Diagram* Secara Umum

Class diagram secara umum yang ditunjukan pada Gambar 5.4 merupakan visualisasi hubungan antar kelas sebagai representasi objek yang menjadi komponen penyusun sistem yang akan dikembangkan. Kelas yang dimodelkan pada diagram diidentifikasi berdasarkan hasil analisis yang telah dilakukan. Pada *class diagram* secara umum terdapat sepuluh kelas dengan setiap kelas memiliki atribut tertentu. Setiap kelas memiliki hubungan dengan kelas lainya. *Class diagram* secara umum akan digunakan sebagai panduan perancangan basis data dan pemodelan *class diagram* untuk kerangka kerja Laravel.



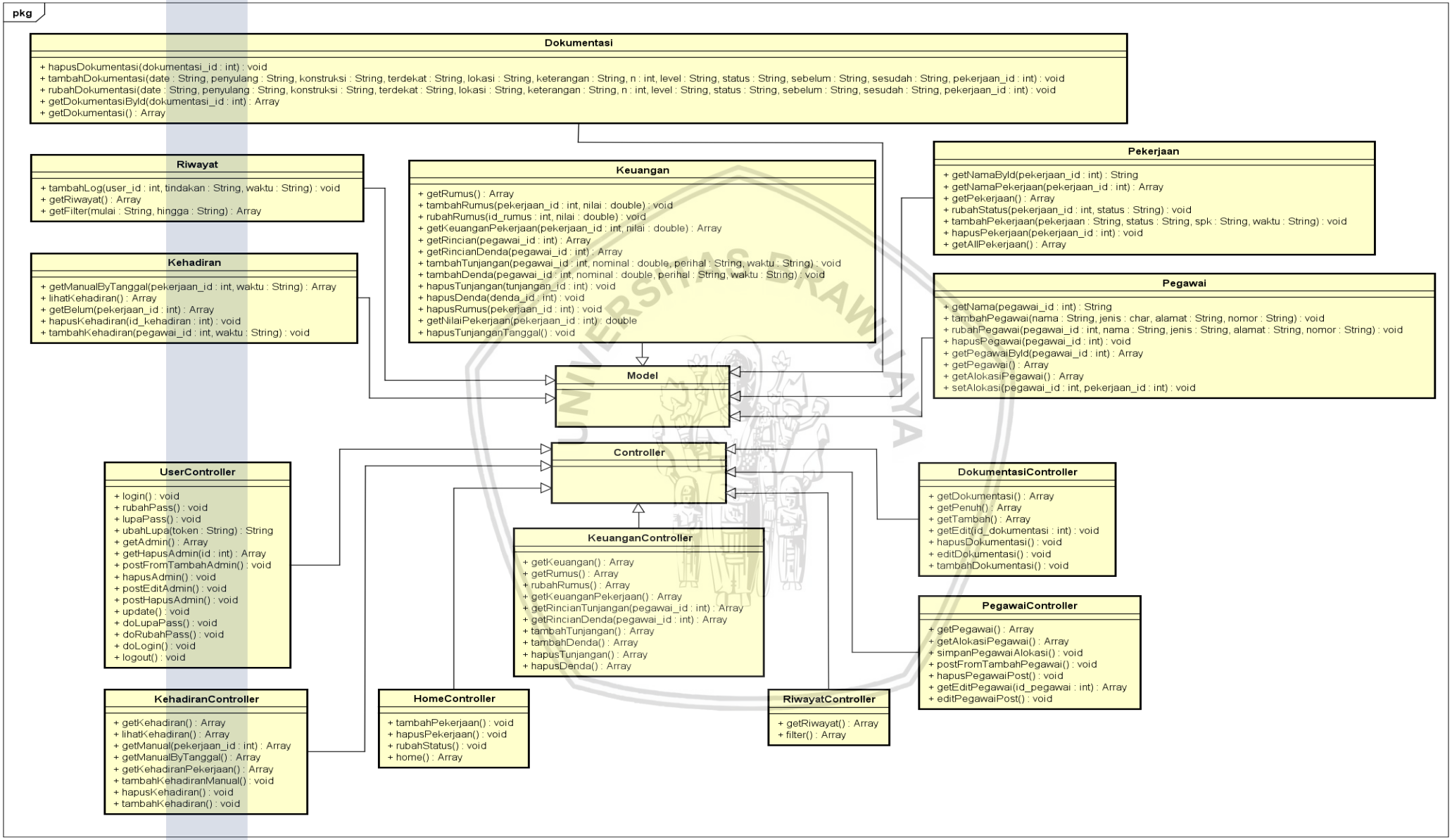
Gambar 5.4 *Class diagram* secara umum

5.2.2 *Class Diagram* Untuk Kerangka Kerja Laravel

Class diagram pada subbab ini merupakan hasil pengembangan *class diagram* secara umum yang telah dibuat. Diagram kelas yang ditunjukan pada Gambar 5.5 dibuat dengan menyesuaikan *class diagram* secara umum, objek-objek yang teridentifikasi pada *sequence diagram*, dan dengan pola MVC pada kerangka kerja pemrogaman PHP Laravel. *Class diagram* untuk kerangka kerja Laravel menggambarkan 7 subkelas *controller* dan 6 subkelas *model.* Fungsi yang teridentifikasi pada setiap subkelas berdasarkan persyaratan fungsional yang telah teridentifikasi pada analisis persyaratan.

Subkelas *controller* secara umum menyediakan fungsi logika yang merepresentasikan kemampuan sistem. Fungsi logika yang merepresentasikan kemampuan sistem untuk mengelola data pegawai ditangani oleh fungsi yang terdapat pada PegawaiController. Fungsi logika yang merepresentasikan kemampuan sistem untuk mengelola data dokumentasi ditangani oleh fungsi pada DokumentasiController. Fungsi logika yang merepresentasikan kemampuan sistem untuk mengelola data keuangan dan melakukan perhitungan keuangan ditangani oleh KeuanganController.

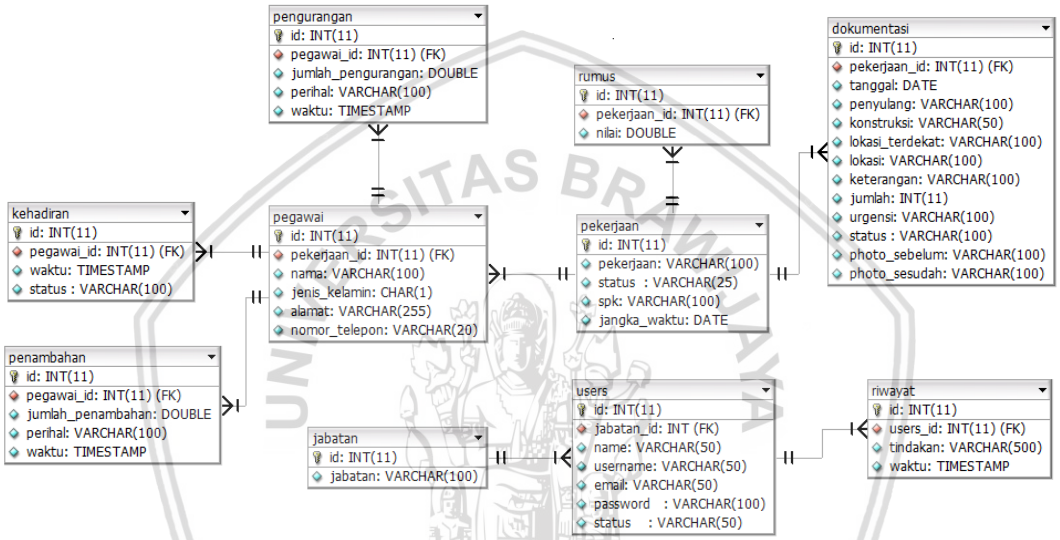
Subkelas *model* secara umum menyediakan fungsi untuk melakukan akses data yang tersimpan pada basis data. Fungsi akses data pegawai ditangani oleh subkelas model Pegawai. Fungsi akses data dokumentasi ditangani oleh subkelas model Dokumentasi. Fungsi untuk akses data keuangan ditangani oleh subkelas model Keuangan.



Gambar 5.5 *Class diagram* untuk kerangka kerja laravel

5.3 Perancangan Basis Data

Perancangan basis data yang terdapat pada Gambar 5.6 dibuat berdasarkan hasil pemodelan *class diagram* secara umum. Setiap kelas pada *class diagram* acuan direpresentasikan oleh tabel yang terdapat pada PDM, sedangkan untuk atribut direpresentasikan dengan nama-nama kolom yang terdapat pada setiap tabel. Hubungan yang terdapat pada PDM dibuat berdasarkan hubungan antar kelas yang terdapat pada *class diagram* secara umum. Dari perancangan basis data didapatkan 10 tabel yang terdiri dari tabel pekerjaan, dokumentasi, pegawai, kehadiran, pengurangan, penambahan, users, rumus, jabatan, dan riwayat.



Gambar 5.6 *Physical data model*

5.3.1 Tabel Pekerjaan

Nama Tabel : pekerjaan
Jumlah Kolom : 5
Fungsi : Untuk menyimpan data pekerjaan

Pada Tabel 5.1 dapat dilihat struktur dari tabel pekerjaan yang digunakan dalam pengembangan sistem.

Tabel 5.1 Tabel pekerjaan

No	Nama Kolom	Tipe Data	Lebar	Keterangan
1	id	INT	11	PK
2	pekerjaan	VARCHAR	100	Jenis pekerjaan
3	status	VARCHAR	25	Status pekerjaan
4	spk	VARCHAR	100	Nomor SPK pekerjaan
5	Jangka_waktu	DATE	-	Jangka waktu pekerjaan

5.3.2 Tabel Dokumentasi

Nama Tabel : dokumentasi
Jumlah Kolom : 13
Fungsi : Untuk menyimpan data dokumentasi pekerjaan

Tabel 5.2 merupakan struktur dari tabel dokumentasi yang digunakan dalam pengembangan sistem.

Tabel 5.2 Tabel dokumentasi

No	Nama Kolom	Tipe Data	Lebar	Keterangan
1	id	INT	11	PK
2	pekerjaan_id	INT	11	FK (Pekerjaan)
3	tanggal	DATE	-	Waktu dokumentasi diambil
4	penyulang	VARCHAR	100	Nama penyulang
5	konstruksi	VARCHAR	50	Nama konstruksi
6	lokasi_terdekat	VARCHAR	100	Nama lokasi terdekat pekerjaan
7	lokasi	VARCHAR	100	Lokasi dokumentasi diambil
8	keterangan	VARCHAR	100	keterangan spesifik pekerjaan yang dilakukan
9	jumlah	INT	11	Jumlah pekerjaan yang dikerjakan
10	urgensi	VARCHAR	100	Tingkat urgensi pekerjaan
11	status	VARCHAR	100	status pekerjaan
12	photo_sebelum	VARCHAR	100	*Link* gambar sebelum pekerjaan dilakukan
13	photo_sesudah	VARCHAR	100	*Link* gambar sesudah pekerjaan dilakukan

5.3.3 Tabel Pegawai

Nama Tabel : pegawai
Jumlah Kolom : 6
Fungsi : Untuk menyimpan data pegawai

Pada Tabel 5.3 dapat dilihat struktur dari tabel pegawai yang digunakan dalam pengembangan sistem.

Tabel 5.3 Tabel pegawai

No	Nama Kolom	Tipe Data	Lebar	Keterangan
1	id	INT	11	PK
2	pekerjaan_id	INT	11	FK (Pekerjaan)
3	nama	VARCHAR	100	Nama pegawai
4	jenis_kelamin	CHAR	1	Jenis kelamin pegawai
5	alamat	VARCHAR	255	Alamat pegawai
6	nomor_telepon	VARCHAR	20	Nomor telepon pegawai

5.3.4 Tabel Kehadiran

Nama Tabel : kehadiran
Jumlah Kolom : 4
Fungsi : Untuk menyimpan data kehadiran pegawai

Pada Tabel 5.4 dapat dilihat struktur dari tabel kehadiran yang digunakan dalam pengembangan sistem.

Tabel 5.4 Tabel kehadiran

No	Nama Kolom	Tipe Data	Lebar	Keterangan
1	id	INT	11	PK
2	pegawai_id	INT	11	FK (Pegawai)
3	waktu	TIMESTAMP	-	Waktu kehadiran pegawai
4	status	VARCHAR	100	Status kehadiran pegawai

5.3.5 Tabel Pengurangan

Nama Tabel : pengurangan
Jumlah Kolom : 5
Fungsi : Untuk menyimpan data pengurangan keuangan

Struktur dari tabel pengurangan yang digunakan dalam pengembangan sistem dapat ditunjukan pada Tabel 5.5.

Tabel 5.5 Tabel pengurangan

No	Nama Kolom	Tipe Data	Lebar	Keterangan
1	id	INT	11	PK
2	pegawai_id	INT	11	FK (Pegawai)
3	jumlah_pengurangan	DOUBLE	-	Pengurangan keuangan pegawai

4	perihal	VARCHAR	100	Ketarangan pengurangan keuangan
5	waktu	TIMESTAMP	-	Waktu pengurangan terjadi

5.3.6 Tabel Penambahan

Nama Tabel : penambahan
Jumlah Kolom : 5
Fungsi : Untuk menyimpan data penambahan keuangan

Pada Tabel 5.6 dapat dilihat struktur dari tabel penambahan yang digunakan dalam pengembangan sistem.

Tabel 5.6 Tabel penambahan

No	Nama Kolom	Tipe Data	Lebar	Keterangan
1	id	INT	11	PK
2	pegawai_id	INT	11	FK(Pegawai)
3	jumlah_penamabahan	DOUBLE	-	Penambahan keuangan pegawai
4	perihal	VARCHAR	100	Keterangan penambahan keuangan
5	waktu	TIMESTAMP	-	Waktu penambahan diberikan

5.3.7 Tabel *Users*

Nama Tabel : *users*
Jumlah Kolom : 7
Fungsi : Untuk menyimpan data *users* (pengguna) sistem

Pada Tabel 5.7 dapat dilihat struktur dari tabel *users* yang digunakan dalam pengembangan sistem.

Tabel 5.7 Tabel *users*

No	Nama Kolom	Tipe Data	Lebar	Keterangan
1	id	INT	11	PK
2	jabatan_id	INT	11	FK(Jabatan)
3	name	VARCHAR	50	Nama pengguna sistem
4	username	VARCHAR	50	*Username* dari pengguna
5	email	VARCHAR	50	*Email* akun pengguna
6	password	VARCHAR	100	*Password* akun pengguna

7	status	VARCHAR	50	Status akun pengguna sistem

5.3.8 Tabel Riwayat

Nama Tabel : riwayat
Jumlah Kolom : 4
Fungsi : Untuk menyimpan data riwayat sistem

Pada Tabel 5.8 dapat dilihat struktur dari tabel riwayat yang digunakan dalam pengembangan sistem.

Tabel 5.8 Tabel riwayat

No	Nama Kolom	Tipe Data	Lebar	Keterangan
1	id	INT	11	PK
2	users_id	INT	11	FK(Users)
3	tindakan	VARCHAR	500	Tindakan yang dilakukan users
4	waktu	TIMESTAMP	-	Waktu tindakan terjadi

5.3.9 Tabel Rumus

Nama Tabel : rumus
Jumlah Kolom : 3
Fungsi : Untuk menyimpan data rumus perhitungan keuangan

Pada Tabel 5.9 dapat dilihat struktur dari tabel rumus yang digunakan dalam pengembangan sistem.

Tabel 5.9 Tabel rumus

No	Nama Kolom	Tipe Data	Lebar	Keterangan
1	id	INT	11	PK
2	pekerjaan_id	INT	11	FK(pekerjaan)
3	nilai	DOUBLE	-	Nilai acuan perhitungan keuangan

5.3.10 Tabel Jabatan

Nama Tabel : jabatan
Jumlah Kolom : 2
Fungsi : Untuk menyimpan data jabatan pegawai

Pada Tabel 5.10 dapat dilihat struktur dari tabel jabatan yang digunakan dalam pengembangan sistem.

Tabel 5.10 Tabel jabatan

No	Nama Kolom	Tipe Data	Lebar	Keterangan

1	id	INT	11	PK
2	jabatan	VARCHAR	100	Nama jabatan

5.4 Perancangan Algoritme

Perancangan algoritme merupakan suatu proses yang dilakukan untuk menentukan dan menyusun serangkaian operasi logika sebagai konsep pembangunan fungsi-fungsi sistem yang akan dikembangkan. Hasil perancangan algoritme dalam penelitian ini didokumentasikan ke dalam bentuk *pseducode*. Hasil perancangan algoritme akan menjadi acuan dalam pengembangan sistem informasi khususnya pada tahap implementasi algoritme. Beberapa perancangan algoritme yang didokumentasikan pada penelitian ini adalah perancangan algoritme pada proses mengalokasi pegawai dalam *use case* mengelola pegawai, menambah data dokumentasi pekerjaan pada *use case* mengelola dokumentasi, dan menampilkan data keuangan pada *use case* mengelola keuangan.

5.4.1 Mengalokasi Pegawai

Pada bagian subbab ini berisi rancangan algoritme dari operasi simpanPegawaiAlokasi pada *class diagram* yang disediakan oleh sistem kepada pengguna pada saat menggunakan sistem untuk mengelola data pegawai. *Pseducode* yang terdapat pada Tabel 5.11 merupakan rancangan algoritme operasi mengalokasi pegawai ke dalam pekerjaan. Rancangan algoritme dimulai dengan mengambil data pekerjaan dan data pegawai terpilih dari *form.* Pegawai terpilih selanjutnya dihitung untuk mengetahui jumlah pegawai.

Jika hasil perhitungan menunjukan ada pegawai yang dipilih oleh pengguna, maka setiap pegawai tersebut akan dimasukan ke dalam pekerjaan. Selanjutnya dijalankan operasi untuk menambah riwayat sistem dan dibuat pesan pada *session* sebagai notifikasi. Terakhir sistem akan mengalihkan halaman ke halaman semula.

Jika tidak ada pegawai terpilih maka tidak ada aktivitas memasukan pegawai ke dalam pekerjaan sehingga sistem hanya perlu membuat pesan notifikasi pada sistem dan mengalihkan halaman ke halaman semula.

Tabel 5.11 Mengalokasi pegawai

No	***Pseducode***
1	Mulai
2	Set pekerjaan = ambil pekerjaan dari form
3	Set id pegawai = ambil id pegawai yang dipilih dari form
4	Count jumlah pegawai yang dipilih
5	Jika jumlah pegawai terpilih lebih dari 0
6	Lakukan perulangan untuk setiap pegawai terpilih

7	Jalankan fungsi untuk mengalokasi pegawai terpilih ke dalam pekerjaan
8	Jalankan fungsi untuk menambah riwayat
9	Akhiri perulangan
10	Set pesan notifikasi pada session
11	Beralih ke tampilan alokasi pegawai
12	Lainya
13	Set pesan notifikasi pada session untuk tidak ada pegawai terpilih
14	Beralih ke tampilan alokasi pegawai
15	Akhiri pengkondisian
16	Selesai

5.4.2 Menambah Data Dokumentasi Pekerjaan

Pada subbab ini berisi rancangan algoritme dari operasi tambahDokumentasi pada *class diagram* yang disediakaan oleh sistem kepada pengguna pada saat menggunakan sistem untuk mengelola dokumentasi pekerjaan. *Pseducode* yang terdapat pada Tabel 5.12 merupakan rancangan algoritme operasi untuk menambah data dokumentasi pekerjaan.

Perancangan algoritme dimulai dengan penyimpanan gambar dan data dokumentasi ke dalam sistem. Selanjutnya membuat pesan pada *session* sebagai notifikasi dan terakhir mengarahkan kembali sistem ke halaman menambah data dokumentasi.

Tabel 5.12 Menambah data dokumentasi pekerjaan

No	***Pseducode***
1	Mulai
2	Set alamat tujuan gambar = alamat penyimpanan lokal
3	Jika pengguna memasukan gambar sebelum pekerjaan dikerjakan
4	Set gambar = gambar sebelum pekerjaan dilakukan
5	Set nama gambar = sebelum_6 string acak_nama gambar
6	Jalankan fungsi memindah gambar ke alamat tujuan gambar
7	Set alamat gambar basis data = nama gambar
8	Lainya
9	Set alamat gambar basis data = nama gambar *default*
10	Akhiri pengkondisian
11	Jika pengguna memasukan gambar sesudah pekerjaan dikerjakan

12	Set gambar = gambar sesudah pekerjaan dilakukan
13	Set nama gambar = sesudah_6 string acak_nama gambar
14	Jalankan fungsi memindah gambar ke alamat tujuan gambar
15	Set alamat gambar basis data = nama gambar
16	Lainya
17	Set alamat gambar basis data = nama gambar *default*
18	Akhiri pengkondisian
19	Jalankan fungsi menambahkan dokumentasi
20	jalankan fungsi menambahkan riwayat
21	Set pesan notifikasi pada session
22	Kembali ke tampilan sebelumnya membawa pesan notifikasi session
23	Selesai

5.4.3 Menampilkan Data Keuangan

Subbab ini berisi rancangan algoritme dari operasi getKeuangan pada *class diagram* yang disediakan oleh sistem kepada pengguna pada saat pengguna menggunakan sistem untuk mengelola data keuangan. Rancangan algoritme yang ditunjukan pada Tabel 5.13 merupakan perancangan algoritme proses untuk menampilkan data keuangan.

Perancangan algoritme dimulai dengan mengambil data pekerjaan, dan rumus perhitungan untuk pekerjaan tersebut. Selanjutnya dilakukan pengambilan keuangan berdasarkan hasil perhitungan menggunakan rumus yang telah diambil sebelumnya.

Tabel 5.13 Menampilkan data keuangan

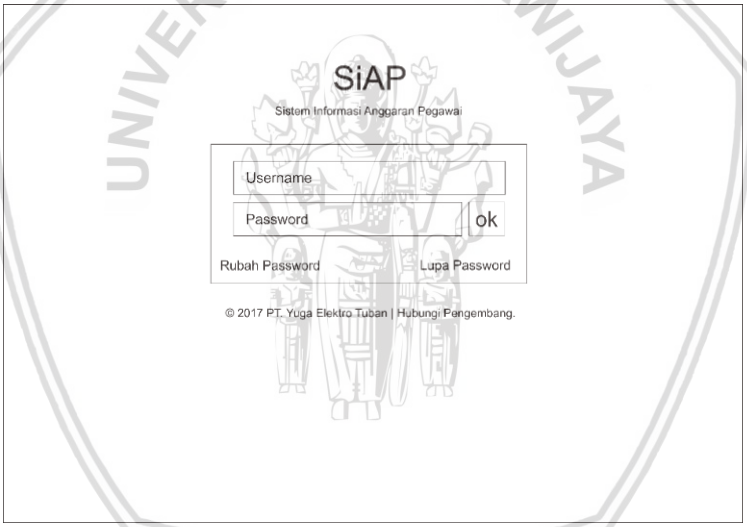
No	*Pseducode*
1	Mulai
2	Set id pekerjaan = ambil id pekerjaan dari *form*
3	Set nama pekerjaan = Jalankan fungsi mengambil nama pekerjaan
4	Set nilai rumus = Jalankan fungsi mengambil rumus pekerjaan
5	Jalankan fungsi mengambil perhitungan keuangan pekerjaan
6	Beralih ke tampilan keuangan pekerjaan
7	Selesai

5.5 Perancangan Antarmuka Pengguna

Pada subbab perancangan antarmuka pengguna menjelaskan beberapa rancangan tampilan antarmuka pengguna sistem informasi yang akan dikembangkan dalam penelitian. Beberapa rancangan antarmuka pengguna tersebut di antaranya adalah antarmuka *login*, antarmuka beranda, antarmuka mengalokasi pegawai, antarmuka menambah dokumentasi, antarmuka menambah kehadiran, dan antarmuka melihat keuangan.

5.5.1 Antarmuka Pengguna *Login*

Antarmuka *login* menampilkan *form* yang dapat diisi oleh pengguna sebelum melakukan otentikasi identitas untuk dapat mengakses informasi melalui sistem berdasarkan hak aksesnya. Komponen dari halaman ini diantaranya adalah logo dan nama sistem informasi, kolom untuk memasukan *username*, kolom untuk *password*, dan tombol untuk menjalankan fungsi otentikasi pengguna. Terdapat juga dua tombol lain untuk mengakses fungsi merubah *password*, dan memulihkan *password* yang terlupa.



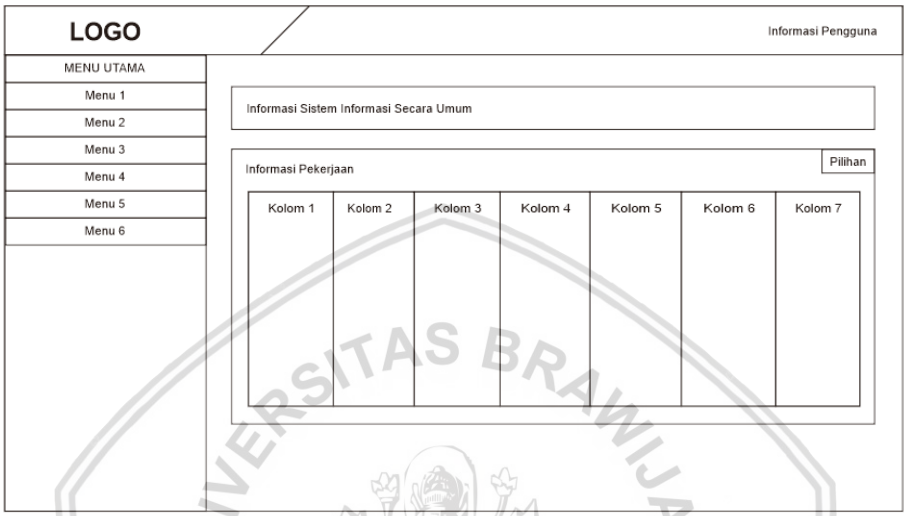
Gambar 5.7 Antarmuka pengguna *login*

5.5.2 Antarmuka Pengguna Beranda

Rancangan antarmuka beranda merupakan tampilan yang akan dilihat oleh pengguna pada saat setelah melakukan otentikasi pengguna. Pada rancangan antarmuka beranda terdapat tiga kelompok komponen yaitu persegi panjang atas, persegi panjang sisi kiri dan persegi panjang sisi kanan.

Pada sisi atas merupakan *header* yang berfungsi untuk menunjukan informasi mengenai pengguna yang terotentifikasi oleh sistem dan logo serta nama dari sistem informasi. Pada sisi kiri merupakan *sidebar* yang berfungsi untuk menampilkan beberapa menu sistem. Pada rancangan antarmuka beranda maka menu sistem beranda akan terpilih oleh sistem.

Kelompok komponen yang terdapat di dalam sisi kanan menampilkan daftar pekerjaan yang telah dimasukan ke dalam sistem. Kolom-kolom yang ditampilkan dalam informasi pekerjaan meliputi id pekerjaan, nama pekerjaan, status pekerjaan, spk pekerjaan, dan jangka waktu pekerjaan. Pada bagian atas sisi kanan terdapat tombol pilihan yang bisa digunakan untuk membuka menu tambahan berupa menu tambah pekerjaan, hapus pekerjaan, dan rubah status pekerjaan.

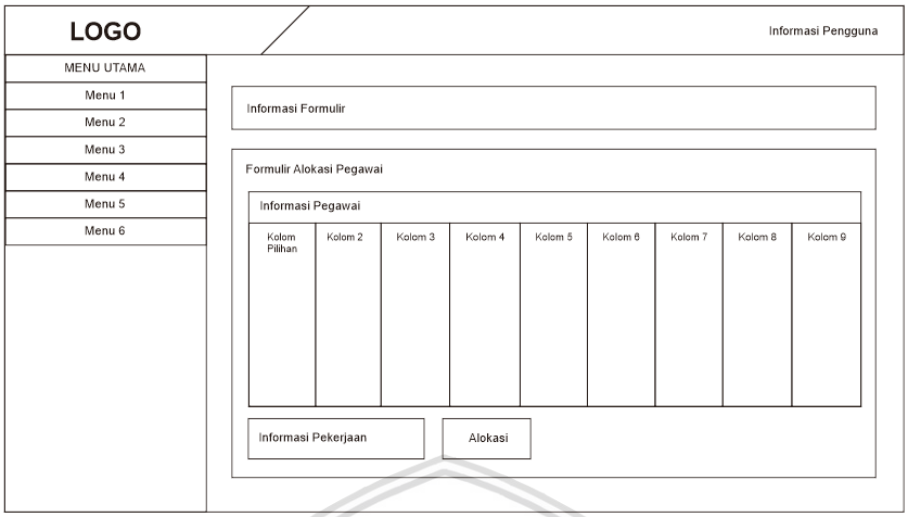


Gambar 5.8 Antarmuka pengguna beranda

5.5.3 Antarmuka Pengguna Mengalokasi Pegawai

Rancangan antarmuka pengguna mengalokasi pegawai akan dilihat oleh pengguna pada saat pengguna menggunakan sistem untuk melakukan alokasi pegawai ke dalam pekerjaan. Antarmuka mengalokasi pegawai dibagi menjadi tiga bagian komponen. Kelompok komponen yang terdapat pada persegi panjang sisi atas, kelompok komponen pada persegi panjang sisi kiri, dan sisi kanan.

Pada kelompok komponen sisi kiri merupakan *sidebar* yang dgunakan untuk menampilkan beberapa menu sistem. Pada antarmuka mengalokasi pegawai maka menu alokasi pegawai akan terpilih oleh sistem. Pada kelompok komponen sisi atas digunakan sebagai *header* untuk menampilkan informasi pengguna yang telah melakukan otentikasi dan logo serta nama sistem informasi.



Gambar 5.9 Antarmuka pengguna mengalokasi pegawai

Pada kelompok komponen sisi kanan berisi informasi formulir dan formulir alokasi pegawai. Informasi formulir berisi penjelasan singkat bagaimana pengguna menggunakan formulir untuk melakukan alokasi pegawai. Pada formulir alokasi pegawai berisi informasi pegawai yang dimasukan ke kolom-kolom di dalam sebuah tabel. Kolom pilihan digunakan untuk memilih pegawai yang akan dialokasikan ke dalam pekerjaan. Kolom lainya berisi informasi pegawai yang dapat berupa nama pegawai dan pekerjaan pegawai. Pada bagian bawah formulir terdapat informasi pekerjaan sebagai tujuan alokasi pegawai dan tombol alokasi untuk melakukan fungsi alokasi pegawai.

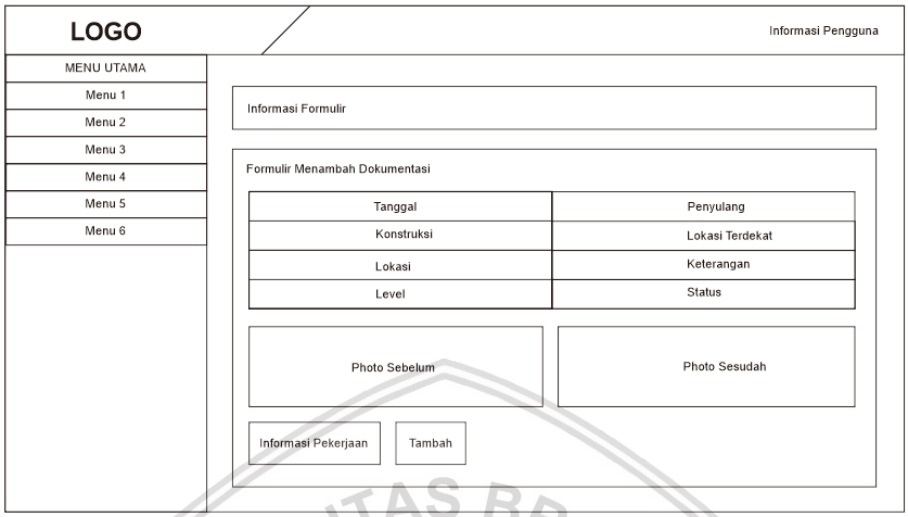
5.5.4 Antarmuka Pengguna Menambah Dokumentasi

Rancangan antarmuka pengguna menambah dokumentasi merupakan antarmuka yang akan ditampilkan kepada pengguna ketika pengguna menggunakan sistem untuk menambah data dokumentasi pekerjaan. Rancangan antarmuka menambah dokumentasi dibagi ke dalam tiga kelompok komponen yang digambarkan dengan tiga buah bangun persegi panjang. Kelompok komponen dibagi menjadi kelompok sisi atas, sisi kiri, dan sisi kanan.

Pada kelompok komponen sisi kiri merupakan *sidebar* yang digunakan untuk menampilkan beberapa menu sistem. Pada antarmuka menambah dokumentasi maka menu tambah dokumentasi akan dipilih oleh sistem. Pada kelompok sisi atas merupakan *header* yang digunakan untuk menampilkan logo dan nama sistem, serta informasi pengguna yang melakukan otentikasi.

Pada sisi kanan terdapat informasi formulir dan formulir menambah dokumentasi. Informasi formulir berisi penjelasan singkat tentang bagaimana pengguna mengisi formulir. Pada formulir menambah dokumentasi pekerjaan terdapat beberapa data yang harus dilengkapi oleh pengguna diantaranya adalah data tanggal, penyulang, konstruksi, lokasi terdekat, lokasi, keterangan tambahan, level pekerjaan, dan status pekerjaan. Pada sisi bawah formulir terdapat dua buah photo untuk dimasukan yaitu photo sebelum dan sesudah pekerjaan dilakukan,

informasi pekerjaan, dan sebuah tombol yang digunakan untuk menjalankan fungsi menambah dokumentasi pekerjaan.



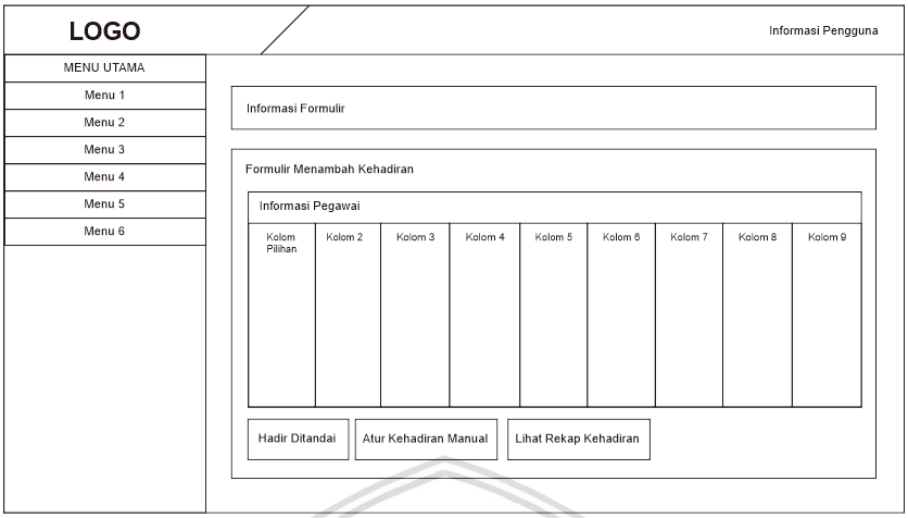
Gambar 5.10 Antarmuka pengguna menambah dokumentasi

5.5.5 Antarmuka Pengguna Menambah Kehadiran

Rancangan antarmuka pengguna menambah kehadiran menunjukan tampilan yang akan dilihat oleh pengguna ketika pengguna menggunakan sistem untuk menambah kehadiran pegawai. Pada Gambar 5.11 menunjukan tiga kelompok komponen antarmuka yang berada pada persegi panjang sisi atas, kiri, dan sisi kanan.

Pada kelompok komponen sisi atas merupakan *header* yang digunakan untuk menampilkan logo dan informasi pengguna yang terotentikasi oleh sistem. Pada sisi kiri merupakan *sidebar* yang digunakan untuk menampilkan beberapa menu sistem. Pada antarmuka menambah kehadiran, menu tambah kehadiran akan dipilih oleh sistem di sisi kiri secara otomatis.

Pada sisi kanan terdapat beberapa komponen diantaranya adalah informasi formulir dan formulir menambah kehadiran. Informasi formulir digunakan untuk memberikan penjelasan singkat tentang formulir yang harus diisi oleh pengguna. Pada formulir menambah kehadiran terdapat informasi pegawai yang dimasukan ke dalam kolom-kolom seperti nama pegawai, pekerjaan pegawai, dan status kehadiran pegawai. Pada sisi kiri informasi pegawai juga terdapat kolom pilihan untuk memilih pegawai yang akan ditambahkan kehadiranya. Pada bagian bawah terdapat tombol hadir ditandai untuk menjalankan fungsi menambah kehadiran, tombol atur kehadiran manual untuk mengatur kehadiran berdasarkan waktu, dan tombol lihat rekap kehadiran untuk melihat data kehadiran seluruh pegawai.



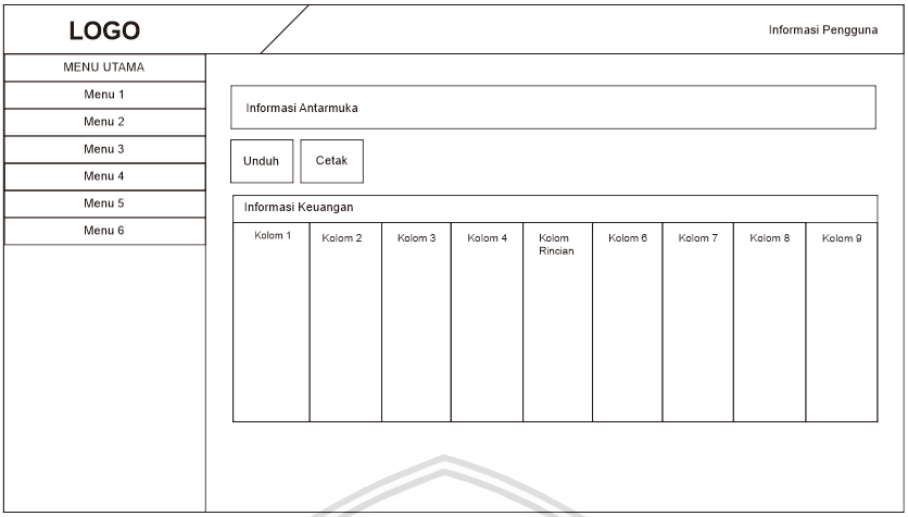
Gambar 5.11 Antarmuka pengguna menambah kehadiran

5.5.6 Antarmuka Pengguna Melihat Keuangan

Rancangan antarmuka melihat keuangan yang ditunjukan pada Gambar 5.12 merupakan rancangan dari sistem yang akan ditampilkan kepada pengguna ketika pengguna menggunakan sistem untuk melihat keuangan. Pada rancangan antarmuka melihat keuangan terdapat tiga kelompok komponen yang berada di dalam persegi sisi atas, persegi sisi kiri, dan persegi sisi kanan.

Pada persegi sisi kiri merupakan *sidebar* yang digunakan untuk menampilkan beberapa menu yang dimiliki sistem. Ketika antarmuka melihat keuangan ditampilkan maka pada *sidebar* akan dipilih menu lihat keuangan. Pada sisi atas merupakan *header* yang berfungsi untuk menampilkan informasi pengguna serta logo dan nama sistem.

Beberapa kelompok komponen yang terdapat pada sisi kanan terdiri dari informasi antarmuka untuk memberikan penjelasan singkat tentang antarmuka yang ditampilkan, dua buah tombol untuk mengunduh dan mencetak data keuangan, dan informasi keuangan. Informasi keuangan ditampilkan kedalam bentuk tabel dengan beberapa kolom di dalamnya. Kolom rincian berfungsi untuk melihat rincian keuangan, sedangkan kolom lainya berisi informasi keuangan seperti nama pegawai, jumlah kehadiran, gaji pokok, penambahan keuangan, pengurangan keuangan, rincian, total pengurangan, dan total pendapatan.



Gambar 5.12 Antarmuka pengguna melihat keuangan

BAB 6 IMPLEMENTASI

Bab ini menjelaskan kebutuhan implementasi dan hasil implementasi sistem informasi yang dikembangkan di dalam penelitian ini. Beberapa hal yang dijelaskan di dalam bagian ini di antaranya adalah spesifikasi lingkungan implementasi, spesifikasi lingkungan *deployment*, beberapa kode implementasi algoritme, dan beberapa antarmuka pengguna sistem informasi yang dikembangkan. Implementasi dilakukan berdasarkan hasil perancangan yang telah dijelaskan pada bab perancangan.

6.1 Spesifikasi Lingkungan Implementasi

Subbab ini menjelaskan spesifikasi perangkat keras dan perangkat lunak yang digunakan selama pengembangan sistem informasi. Perangkat keras yang digunakan pada proses pengembangan sistem informasi berupa sebuah unit laptop dengan spesifikasi yang dapat dijelaskan pada Tabel 6.1. Terdapat pula spesifikasi perangkat lunak yang digunakan dalam pengembangan sistem informasi yang dapat ditunjukan pada Tabel 6.2.

Tabel 6.1 Spesifikasi perangkat keras

Unit Komputasi	Laptop ASUS A450L
Processor	Intel® Core™ i5-4200U CPU @ 1.60GHz 2.30GHz
Kapasitas Memori (RAM)	8 GB
Kapasitas Penyimpanan	500 GB
Maksimal Resolusi Layar	1366px x 768px
Kartu Grafis	Nvidia GeForce 820M

Tabel 6.2 Spesifikasi perangkat lunak

Sistem Operasi	Windows 10 Pro (64-bit)
Bahasa Pemrogaman	PHP, HTML, CSS, Javascript, SQL
Perangkat Lunak Pendukung	DBDesigner Astah Sublime Ms Word 2016 Google Chrome phpMyAdmin
***Library* Pendukung**	Jquery Laravelcollective html IziToast

6.2 Spesifikasi Lingkungan *Deployment*

Subbab ini menjelaskan spesifikasi server yang harus terpenuhi sebelum memasang sistem informasi ke dalam server. Adapun sistem yang dikembangkan menggunakan kerangka pemrogaman PHP Laravel 4.2 memerlukan spesifikasi server minimum yang dapat dijelaskan pada Tabel 6.3.

Tabel 6.3 Spesifikasi server

Dependensi	Composer v1.6.4
Versi PHP	>= 5.4
Ekstensi	Mcrypt PHP Extension
Kapasitas Penyimpanan	160 GB

6.3 Implementasi Algoritme

Implementasi algoritme sistem informasi yang dijelaskan pada subbab ini berdasar pada *pseducode* dalam perancangan algoritme yang telah dibuat pada tahap perancangan. Hasil implementasi algoritme dikelompokan ke dalam beberapa bagian yang berkaitan dengan proses mengelola pegawai, mengelola dokumentasi, dan mengelola keuangan.

6.3.1 Mengelola pegawai

Kode program yang terdapat pada Tabel 6.4 dan Tabel 6.5 merupakan implementasi algoritme fungsi untuk mengalokasi pegawai ke dalam pekerjaan yang merupakan bagian dari layanan pengelolaan pegawai yang disediakan oleh sistem. Implementasi algoritme mengalokasi pegawai dibuat berdasarkan hasil rancangan algoritme mengalokasi pegawai pada bab perancangan.

Tabel 6.4 Implementasi algoritme mengalokasi pegawai (*controller*)

PegawaiController.php	
1	public function simpanPegawaiAlokasi(){
2	$pekerjaan_id = Input::get('pekerjaan');
3	$id_pegawai = Input::get('chkboxName');
4	$max = count($id_pegawai);
5	if ($max > 0) {
6	for ($i = 0; $i < $max; $i++){
7	$pegawai_id = $id_pegawai[$i];
8	pegawai::setAlokasi($pegawai_id,$pekerjaan_id);
9	$namaPekerjaan = pekerjaan::getNamaById($pekerjaan_id);

10	$namaPegawai = pegawai::getNama($pegawai_id);
11	riwayat::tambahLog(Auth::user()->id,'Mengalokasikan pegawai. (Nama pegawai : '.$
12	namaPegawai.', Pekerjaan : '.$namaPekerjaan.')',date("Y-m-d H:i:s"));
13	}
14	Session::flash('pegawaiAlokasi', ' ');
15	return Redirect::to('pegawaiAlokasi');
16	}
17	else{
18	Session::flash('pegawaiAlokasiTdk', ' ');
19	return Redirect::to('pegawaiAlokasi');
20	}
21	}

Penjelasan singkat dari Tabel 6.4 yaitu:

1. Baris ke-2 sampai ke-4 merupakan inisiasi variabel dari hasil pemanggilan fungsi.
2. Baris ke-5 sampai ke-16 merupakan kondisi jika terdapat pegawai yang dipilih oleh pengguna untuk dialokasi.
3. Baris ke-6 sampai ke-15 merupakan proses alokasi pegawai ke dalam pekerjaan
4. Baris ke-17 sampai ke-20 merupakan kondisi yang dijalankan jika pengguna tidak memilih pegawai untuk dialokasi.

Tabel 6.5 Implementasi algoritme mengalokasi pegawai (*model*)

PegawaiModel.php	
1	function setAlokasi($pegawai_id, $pekerjaan_id){
2	DB::table('pegawai')
3	->where ('id', $pegawai_id)
4	->update(array(
5	'pekerjaan_id' => $pekerjaan_id
6	));
7	}}

Penjelasan singkat dari Tabel 6.5 yaitu:

1. Baris ke-1 merupakan deklarasi fungsi mengalokasi pegawai.

2. Baris ke-2 sampai ke-5 merupakan proses pemasukan data ke dalam basis data.

6.3.2 Mengelola Dokumentasi

Implementasi algoritme dalam bentuk kode program yang terdapat pada Tabel 6.6 merupakan implementasi fungsi menambah data dokumentasi pekerjaan yang disediakan oleh sistem untuk memenuhi kebutuhan mengelola data dokumentasi pekerjaan. Implementasi algoritme dilakukan berdasarkan hasil rancangan algoritme menambah data dokumentasi pekerjaan pada bab perancangan.

Tabel 6.6 Implementasi algoritme menambah data dokumentasi pekerjaan (*controller*)

DokumentasiController.php	
1	public function tambahDokumentasi(){
2	$filename ='';
3	$filenameSudah ='';
4	$destinationPath = public_path().'/img/rabas/';
5	if (Input::hasFile('sebelum')) {
6	$file = Input::file('sebelum');
7	$filename = 'sebelum_'.str_random(6).'_'.$file->getClientOriginalName();
8	$upload = $file->move($destinationPath,$filename);
9	$baseName = 'img/rabas/'.$filename;
10	}
11	else{
12	$baseName = 'img/rabas/default.jpg';
13	}
14	if (Input::hasFile('sesudah')) {
15	$file = Input::file('sesudah');
16	$filenameSudah = 'sesudah_'.str_random(6).'_'.$file->getClientOriginalName();
17	$upload = $file->move($destinationPath,$filenameSudah);
18	$baseNameSudah = 'img/rabas/'.$filenameSudah;
19	}
20	else{
21	$baseNameSudah = 'img/rabas/default.jpg';
22	}

23	dokumentasi::tambahDokumentasi(Input::get('date'),Input::get('penyulang'),Input::get('konstruksi'
24	),Input::get('terdekat'),Input::get('lokasi'),Input::get('pohon'),Input::get('n'),Input::get(
25	'level'),Input::get('ket'),$baseName,$baseNameSudah,Input::get('pekerjaan'));
26	riwayat::tambahLog(Auth::user()->id,'Menambah data dokumentasi. (Penyulang : '.Input::get('
27	penyulang').', Lokasi Terdekat : '.Input::get('terdekat').')',date("Y-m-d H:i:s"));
28	Session::flash('rabasAdd', ' ');
29	return Redirect::to('tambahRabas');
30	}

Penjelasan dari Tabel 6.6 yaitu:

1. Baris ke-1 sampai ke-4 merupakan inisiasi variable yang akan digunakan.
2. Baris ke-5 sampai ke-22 merupakan proses pengambilan gambar dokumentasi dan penyimpanan ke dalam penyimpanan lokal dan variabel.
3. Baris ke-23 sampai ke-25 merupakan proses pemanggilan fungsi untuk menyimpan dokumentasi ke dalam basis data
4. Baris ke-26 sampai ke-27 merupakan proses pemanggilan fungsi untuk menyimpan riwayat sistem
5. Baris ke-28 merupakan penyimpanan pesan ke dalam *session*
6. Baris ke-29 merupakan pengalihan halaman

Tabel 6.7 Implementasi algoritme menambah data dokumentasi pekerjaan (*model*)

DokumentasiModel.php	
1	function tambahDokumentasi($date,$penyulang,$konstruksi,$terdekat,$lokasi,$keterangan,$n,$level,$status,$
2	sebelum,$sesudah,$pekerjaan){
3	DB::table('rabas')->insert(
4	array(
5	'tanggal' => $date,
6	'penyulang' => $penyulang,
7	'konstruksi' => $konstruksi,
8	'lokasi_terdekat' => $terdekat,
9	'lokasi' => $lokasi,
10	'keterangan' => $keterangan,

11	'jumlah' => $n,
12	'urgensi' => $level,
13	'status' => $status,
14	'sebelum' => $sebelum,
15	'sesudah' => $sesudah,
16	'pekerjaan_id' => $pekerjaan
17	);}

Penjelasan dari Tabel 6.7 yaitu :

1. Baris ke-1 sampai ke-2 merupakan deklarasi dari fungsi menambah dokumentasi
2. Baris ke-3 merupakan *query* untuk menandakan insert
3. Baris ke-4 sampai ke-16 merupakan penentuan parameter untuk dimasukan ke dalam nama kolom

6.3.3 Mengelola Keuangan

Kode program pada Tabel 6.8 merupakan hasil implementasi algoritme dari fungsi menampilkan data keuangan yang disediakan oleh sistem untuk memenuhi kebutuhan mengelola keuangan. Implementasi algoritme menampilkan data keuangan dilakukan berdasarkan hasil perancangan algoritme menampilkan data keuangan pada bab perancangan.

Tabel 6.8 Implementasi algoritme menampilkan data keuangan (*controller*)

KeuanganController.php	
1	public function getKeuanganPekerjaan(){
2	$pekerjaan_id = Input::get('pekerjaan');
3	$namaPekerjaan = pekerjaan::getNamaPekerjaan($pekerjaan_id);
4	$nilai = keuangan::getNilaiPekerjaan($pekerjaan_id);
5	$keuanganPekerjaan = keuangan::getKeuanganPekerjaan($pekerjaan_id, $nilai);
6	return View::make('keuangan.keuanganPekerjaan',$namaPekerjaan,$keuanganPekerjaan);}

Penjelasan dari Tabel 6.8 yaitu:

1. Baris ke-2 merupakan inisiasi variabel untuk menyimpan id pekerjaan
2. Baris ke-3 merupakan pemanggilan fungsi untuk pengambilan nama pekerjaan
3. Baris ke-4 merupakan pemanggilan fungsi untuk mengambil nilai rumus keuangan pekerjaan

4. Baris ke-5 merupakan pemanggilan fungsi untuk menghitung dan mendapatkan data keuangan pekerjaan
5. Baris ke-6 merupakan pengalihan halaman untuk melihat hasil perhitungan

Tabel 6.9 Implementasi algoritme menampilkan data keuangan (*model*)

KeuanganModel.php	
1	function getKeuanganPekerjaan($pekerjaan_id, $nilai){
2	$keuanganPekerjaan = DB::select("
3	SELECT p.id, p.nama, count(DISTINCT(a.waktu)) masuk, (count(DISTINCT(a.waktu))*$nilai)
4	pokok, COALESCE(sum(DISTINCT(t.tunjangan)),0) tunjangan, COALESCE(sum(DISTINCT(d.denda
5	)),0) denda, ((count(DISTINCT(a.waktu))*$nilai) + COALESCE(sum(DISTINCT(t.tunjangan)),0
6	) - COALESCE(sum(DISTINCT(d.denda)),0))total
7	FROM pegawai p JOIN absensi a
8	ON p.id = a.pegawai_id
9	LEFT JOIN tunjangan t
10	ON t.pegawai_id = p.id
11	LEFT JOIN denda d
12	ON d.pegawai_id = p.id
13	WHERE month(a.waktu) = month(curdate()) AND p.pekerjaan_id = '$pekerjaan_id'
14	GROUP BY p.id
15	");
16	$keuanganPekerjaan = [
17	'keuanganPekerjaan' => $keuanganPekerjaan
18	];
19	return $keuanganPekerjaan;}

Penjelasan dari Tabel 6.9 yaitu:
1. Baris ke-1 merupakan inisiasi dari fungsi dengan dua parameter
2. Baris ke-2 sampai ke-15 merupakan *query* yang digunakan untuk menghitung dan mengambil data keuangan pekerjaan
3. Baris ke-16 sampai ke-18 merupakan pemasukan hasil query ke dalam *array*
4. Baris ke-19 pengembalian *array* kepada *controller*

6.4 Implementasi Antarmuka Pengguna

Pada subbab implementasi antarmuka pengguna dijelaskan beberapa hasil implementasi antarmuka pengguna sistem informasi dalam bentuk gambar. Beberapa gambar yang dijelaskan mengikuti perancangan antarmuka yang telah dijelaskan sebelumnya pada bab perancangan diantaranya adalah antarmuka pengguna *login,* antarmuka pengguna beranda, antarmuka pengguna mengalokasi pegawai, antarmuka pengguna menambah dokumentasi, antarmuka pengguna menambah kehadiran, dan antarmuka pengguna melihat keuangan.

6.4.1 Antarmuka Pengguna *Login*

Antramuka pengguna *login* ditampilkan kepada pengguna pada saat pengguna menggunakan sistem untuk melakukan otentikasi identitasnya. Antarmuka *login* yang ditunjukan pada Gambar 6.1 memiliki beberapa komponen diantaranya judul sistem, nama sistem, kolom untuk memasukan *username* dan *password*, dan tiga buah tombol yang masing-masing berfungsi untuk menjalankan fungsi otentikasi, menjalankan fungsi untuk merubah *password*, dan menjalankan fungsi untuk memulihkan *password.*

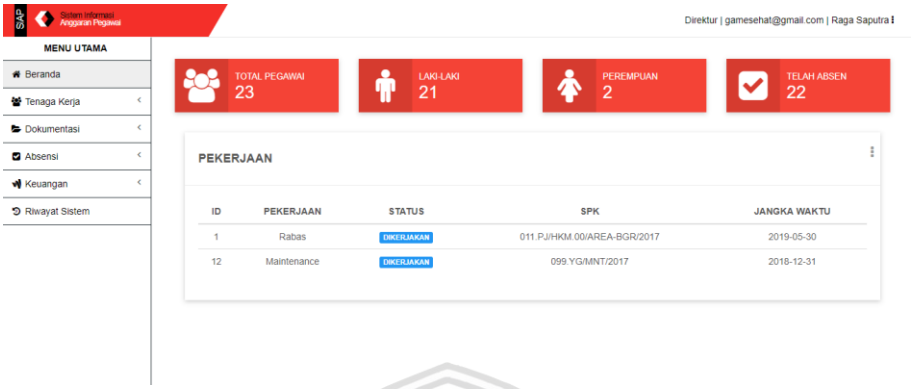
Gambar 6.1 Implementasi antarmuka pengguna *login*

6.4.2 Antarmuka Pengguna Beranda

Antarmuka pengguna beranda ditampilkan kepada pengguna sesaat setelah pengguna sukses melakukan otentikasi ke dalam sistem. Hasil implementasi antarmuka pengguna beranda menunjukan tiga buah kelompok komponen yang dapat dibedakan menjadi sisi atas, sisi kiri, dan sisi kanan.

Pada sisi atas merupakan *header* yang berfungsi untuk menampilkan logo, nama sistem informasi, dan informasi pengguna sistem informasi. Pada sisi kiri terdapat *sidebar* yang berfungsi untuk menampilkan beberapa menu yang dimiliki oleh sistem. Pada sisi kanan yang merupakan bagian utama berisi *dashboard* yang memuat data kehadiran pegawai, dan informasi pekerjaan yang telah disimpan di dalam sistem. Pada sebelah kanan atas informasi pekerjaan juga terdapat tombol

yang dapat digunakan untuk menambah pekerjaan, menghapus pekerjaan, dan merubah status pekerjaan.

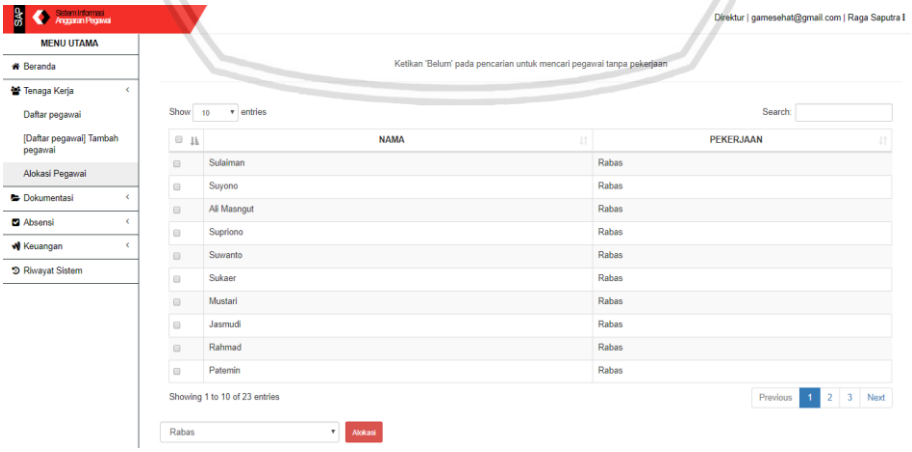


Gambar 6.2 Implementasi antarmuka pengguna beranda

6.4.3 Antarmuka Pengguna Mengalokasi Pegawai

Antarmuka pengguna mengalokasi pegawai ditampilkan kepada pengguna ketika pengguna menggunakan sistem untuk mengalokasi pegawai ke dalam pekerjaan. Antarmuka yang ditunjukan pada Gambar 6.3 tersebut memiliki beberapa komponen.

Pada bagian atas antarmuka terdapat *header* yang berfungsi untuk menampilkan informasi pengguna sistem, logo, dan nama sistem. Terdapat *sidebar* pada sisi sebelah kiri antarmuka pengguna yang menampilkan beberapa menu yang terdapat di dalam sistem. Pada bagian utama sistem terdapat informasi formulir untuk membantu pengguna dalam memahami pengisian formulir alokasi pegawai. Pada bagian utama juga terdapat informasi pegawai yang dapat dipilih untuk dialokasikan ke dalam pekerjaan. Pada bagian bawah terdapat informasi pekerjaan, dan sebuah tombol untuk menjalankan fungsi mengalokasi pegawai.



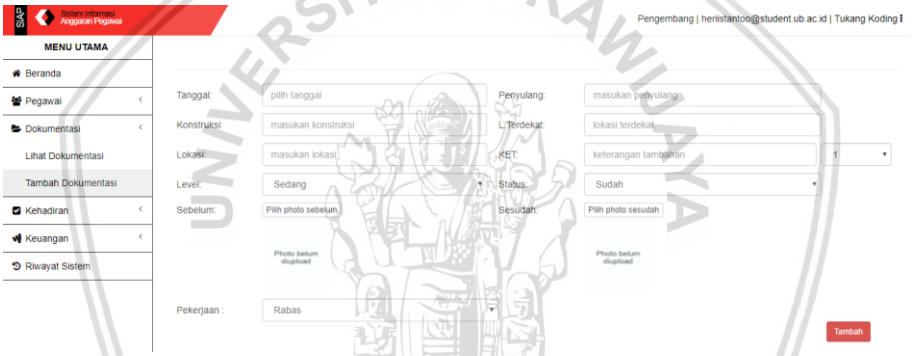
Gambar 6.3 Implementasi antarmuka pengguna mengalokasi pegawai

6.4.4 Antarmuka Pengguna Menambah Dokumentasi

Hasil implementasi antarmuka pengguna menambah dokumentasi yang dapat ditunjukan pada Gambar 6.4 merupakan tampilan yang diberikan oleh sistem kepada pengguna ketika pengguna menggunakan sistem untuk menambah data dokumentasi pekerjaan. Antarmuka ini dibedakan menjadi tiga bagian.

Bagian pertama berupa *header* yang berfungsi untuk menampilkan informasi pengguna yang terotentifikasi oleh sistem, nama sistem informasi, dan logo sistem informasi. Bagian kedua yang terletak di sisi sebelah kiri merupakan *sidebar* yang berfungsi untuk menampilkan menu yang tersedia di dalam sistem.

Bagian terakhir merupakan bagian utama antarmuka pengguna menambah dokumentasi. Pada bagian utama terdapat formulir menambah dokumentasi yang berisi kolom untuk memasukan data dokumentasi, kolom untuk memasukan photo dokumentasi, kolom untuk memilih pekerjaan, dan tombol untuk menjalankan fungsi menambah data dokumentasi.

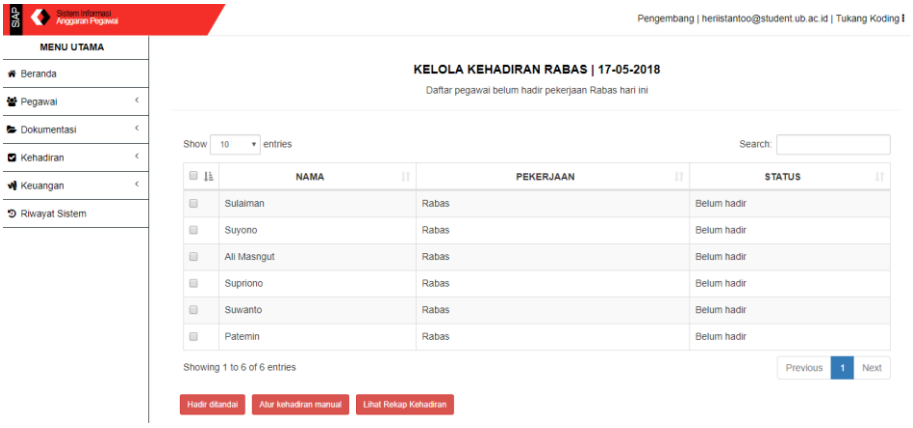


Gambar 6.4 Implementasi antarmuka pengguna menambah dokumentasi

6.4.5 Antarmuka Pengguna Menambah Kehadiran

Antarmuka pengguna menambah kehadiran merupakan tampilkan yang akan dilihat oleh pengguna pada saat pengguna menggunakan sistem untuk menambah kehadiran pengguna. Pada antarmuka yang ditunjukan pada Gambar 6.5 tersebut terdapat *sidebar* yang berfungsi untuk menampilkan beberapa menu yang dimiliki oleh sistem. Terdapat *header* yang berfungsi untuk menampilkan informasi pengguna, judul dan logo sistem.

Pada bagian utama antarmuka di sisi atas terdapat informasi formulir untuk memberikan penjelasan singkat tentang formulir yang harus digunakan oleh pengguna. Pada bagian bawahnya terdapat formulir menambah kehadiran yang dilengkapi dengan kolom pilihan untuk memilih pegawai, nama pegawai, pekerjaan, dan status pegawai. Pada bagian bawah terdapat tiga buah tombol yang masing-masing berfungsi untuk menjalankan fungsi tambah kehadiran, atur kehadiran sesuai waktu, dan melihat rekap kehadiran.



Gambar 6.5 Implementasi antarmuka pengguna menambah kehadiran

6.4.6 Antarmuka Pengguna Melihat Keuangan

Antarmuka pengguna melihat keuangan merupakan tampilan yang dilihat oleh pengguna ketika pengguna menggunakan sistem untuk melihat data hasil perhitungan keuangan. Hasil implementasi antarmuka pengguna melihat keuangan yang dapat dilihat pada Gambar 6.6 menghasilkan komponen *sidebar, header,* dan bagian utama.

Pada bagian *sidebar* terdapat beberapa menu sistem. Pada bagian *header* terdapat informasi pengguna yang terotentifikasi, logo dan nama sistem. Pada bagian utama berfungsi untuk menampilkan informasi singkat mengenai antarmuka untuk mempermudah pengguna dalam memahami antarmuka. Pada bagian bawahnya terdapat dua buah tombol yang berfungsi untuk mengunduh dan mencetak laporan keuangan. Pada bagian paling bawah terdapat informasi keuangan yang disajikan ke dalam bentuk tabel dengan beberapa kolom untuk memperjelas informasi. Kolom tersebut diantaranya kolom nama pegawai, jumlah kehadiran, gaji pokok, total penambahan, total pengurangan, dan total gaji yang didapatkan oleh pegawai.

Gambar 6.6 Implementasi antarmuka pengguna melihat keuangan

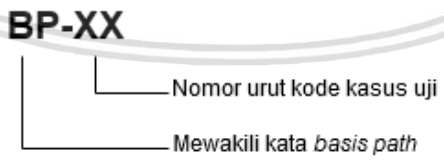
BAB 7 PENGUJIAN

Bab ini menjelaskan pengujian terhadap sistem yang telah dikembangkan pada penelitian. Pengujian dilakukan untuk mengetahui bahwa setiap komponen dari sistem telah sesuai dengan perancangan dan persyaratan yang telah di tetapkan sebelumnya, serta sesuai dengan harapan pengguna. Terdapat tiga jenis pengujian yang akan dilakukan yaitu pengujian *basis path,* pengujian validasi*,* dan pengujian kompatibilitas. Pada tiap pengujian akan disertakan mekanisme pengujian, hasil pengujian, serta kesimpulan pengujian yang telah dilakukan.

7.1 Pengujian *Basis Path*

Teknik pengujian *basis path* yang termasuk ke dalam kategori pengujian *white box* dipilih atas dasar teori Pressman (2010) yang mengatakan bahwa pengujian *white box* yang menguji jalur logis sistem memiliki keunggulan untuk mengetahui kebenaran sistem hingga seratus persen. Namun akan membutuhkan waktu yang lama jika menguji seluruh jalur logis yang terdapat pada sistem. Bahkan untuk program kecil, jumlah jalur logis yang mungkin terjadi bisa sangat besar. Pressman (2010) menyimpulkan bahwa pengujian *white box* cocok dilakukan pada beberapa fungsi yang dirasa penting saja. Teknik pengujian *basis path* sendiri sangat cocok digunakan untuk memperoleh ukuran kompleksitas logis dari sebuah fungsi dan menggunakan ukuran tersebut untuk menentukan kumpulan jalur eksekusi.

Mekanisme pengujian pertama akan dituliskan *pseducode* dari fungsi yang akan diuji berdasarkan hasil implementasi algoritme mengalokasi pegawai dan menambah data dokumentasi pekerjaan bagian *controller*, kemudian akan dihitung *cyclomatic complexity* untuk menentukan jalur *independent path*. Jalur tersebut selanjutnya diujikan pada suatu *test case* sehingga dapat diketahui status validasi antara harapan dan hasil pengujian. Kasus uji pengujian *basis path* memilki kodifikasi sebagai identitas yang dapat ditunjukan pada Gambar 7.1.



Gambar 7.1 Keterangan kodifikasi pengujian *basis path*

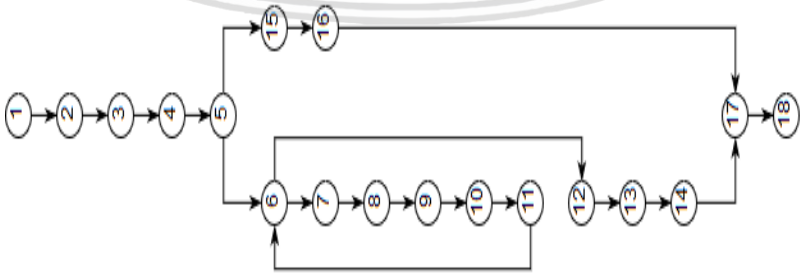
7.1.1 Pengujian *Basis Path* Mengalokasi Pegawai

Pada Tabel 7.1 merupakan *pseducode* dari implementasi algoritme mengalokasi pegawai bagian *controller* yang akan digunakan sebagai acuan dalam menentukan *cyclomatic complexity* dan *independent path.*

Tabel 7.1 *Pseducode* mengalokasi pegawai

No	Pseducode
1	Mulai ----------1
2	Set id pekerjaan = ambil id pekerjaan dari *form* ----------2
3	Set id pegawai = ambil id pegawai dai *form* ----------3
4	Set max = hitung total pegawai yang dipilih ----------4
5	Jika max lebih dari 0 ----------5
6	Lakukan perulangan ----------6
7	Set id = id pegawai index ke n ----------7
8	Jalankan fungsi menambah alokasi pegawai ke pekerjaan ----------8
9	Set nama pekerjaan = jalankan fungsi ambil nama pekerjaan ----------9
10	Set nama pegawai = jalankan fungsi ambil nama pegawai ----------10
11	Jalankan fungsi menambah riwayat ----------11
12	Akhiri perulangan ----------12
13	Set pesan notifikasi pada session ----------13
14	Kembali ke tampilan alokasi pegawai ----------14
15	Lainya
16	Set pesan notifikasi pada session ----------15
17	Kembali ke tampilan alokasi pegawai ----------16
18	Akhiri pengkondisian ----------17
19	Selesai ----------18

Flow Graph yang dihasilkan dari *pseducode* yang terdapat pada Tabel 7.1 dapat dilihat pada gambar 7.2.



Gambar 7.2 *Flow graph* mengalokasi pegawai

Dari *flow graph* pada Gambar 7.2 dapat dilakukan perhitungan *cyclomatic complexity* dengan perhitungan sebagai berikut :

V(G) = Jumlah busur-jumlah simpul+2 = 19-18+2 = 3
V(G) = Jumlah *predicate* node +1 = 2+1 = 3

V(G) = Jumlah region = 3

Dari *cyclomatic complexity* yang diperoleh didapatkan jalur *independent* sejumlah 3. Jalur – jalur tersebut antara lain:

1) 1-2-3-4-5-15-16-17-18
2) 1-2-3-4-5-6-12-13-14-17-18
3) 1-2-3-4-5-6-7-8-9-10-11-12-13-14-17-18

Berdasarkan 3 jalur tersebut didapatkan *test case* yang dapat dijelaskan pada Tabel 7.2 dan Tabel 7.3.

a. Deskripsi Pengujian

Tabel 7.2 Deskripsi pengujian mengalokasi pegawai

Nama proyek	Sistem Informasi Anggaran Pegawai
ID pengujian	BP-1
Fungsi	Fungsi mengalokasi pegawai
Deskripsi	*Test case* ini digunakan menguji fungsi mengalokasi pegawai pada bagian jumlah pegawai yang dipilih.
Perancang	Peneliti
Penguji	Peneliti
Persyaratan Uji	Terhubung ke basis data pegawai dan pekerjaan

b. Prosedur Pengujian

Tabel 7.3 Prosedur pengujian mengalokasi pegawai

No	Skenario pengujian	Data masukan	Hasil diharapkan	Hasil pengujian	Status
1	1. Penguji memilih menu alokasi pegawai 2. Penguji menekan tombol alokasi	Jumlah id pegawai == 0	Keluar notifikasi tidak ada pegawai dipilih	Keluar notifikasi tidak ada pegawai dipilih	Valid
2	1. Kondisi perulangan dirubah menjadi gagal 2. Penguji memilih menu alokasi pegawai 3. Penguji memilih pegawai 4. Penguji menekan tombol alokasi	Jumlah id pegawai > 0 dan perulangan gagal	Keluar notifikasi Alokasi berhasil diberikan, namun data tidak berubah	Keluar notifikasi Alokasi berhasil diberikan, namun data tidak berubah	Valid

3	1. Penguji memilih menu alokasi pegawai 2. Penguji memilih pegawai 3. Penguji menekan tombol alokasi	Jumlah id > 0 dan perulangan sukses	Data pekerjaan pegawai pada tabel alokasi pegawai berubah	Data pekerjaan pegawai pada tabel alokasi pegawai berubah	Valid

7.1.2 Pengujian *Basis Path* Menambah Data Dokumentasi Pekerjaan

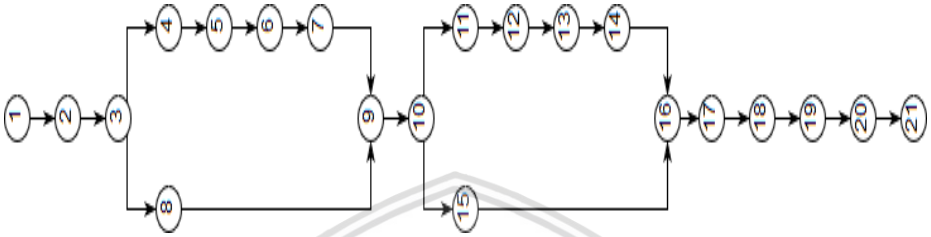
Pada Tabel 7.4 adalah *pseducode* dari implementasi algoritme menambah data dokumentasi pekerjaan bagian *controller* yang akan digunakan sebagai acuan dalam menentukan *cyclomatic complexity* dan *independent path.*

Tabel 7.4 *Pseducode* menambah data dokumentasi pekerjaan

No	Pseducode
1	Mulai ----------1
2	Set alamat tujuan gambar = alamat penyimpanan lokal ----------2
3	Jika pengguna memasukan gambar sebelum pekerjaan dikerjakan ----------3
4	Set gambar = gambar sebelum pekerjaan dilakukan ----------4
5	Set nama gambar = sebelum_6 string acak_nama gambar ----------5
6	Jalankan fungsi memindah gambar ke alamat tujuan gambar ----------6
7	Set alamat gambar basis data = nama gambar ----------7
8	Lainya
9	Set alamat gambar basis data = nama gambar default ----------8
10	Akhiri pengkondisian ----------9
11	Jika pengguna memasukan gambar sesudah pekerjaan dikerjakan ----------10
12	Set gambar = gambar sesudah pekerjaan dilakukan ----------11
13	Set nama gambar = sesudah_6 string acak_nama gambar ----------12
14	Jalankan fungsi memindah gambar ke alamat tujuan gambar ---------13
15	Set alamat gambar basis data = nama gambar ----------14
16	Lainya
17	Set alamat gambar basis data = nama gambar default ----------15
18	Akhiri pengkondisian ----------16
19	Jalankan fungsi menambahkan dokumentasi ----------17
20	jalankan fungsi menambahkan riwayat ----------18
21	Set pesan notifikasi pada session ----------19

22	Kembali ke tampilan sebelumnya dengan membawa pesan notifikasi session ----------20
23	Selesai ----------21

Flow Graph yang dihasilkan dari *pseducode* yang terdapat pada Tabel 7.4 dapat dilihat pada Gambar 7.3



Gambar 7.3 *Flow graph* menambah data dokumentasi pekerjaan

Dari *flow graph* pada Gambar 7.3 dapat dilakukan perhitungan *cyclomatic complexity* dengan perhitungan sebagai berikut :

V(G) = Jumlah busur-jumlah simpul+2 = 22-21+2 = 3
V(G) = Jumlah predicate node +1 = 2+1 = 3
V(G) = Jumlah region = 3

Dari *cyclomatic complexity* yang diperoleh didapatkan jalur *independent* sejumlah 3. Jalur – jalur tersebut antara lain:

1) 1-2-3-8-9-10-15-16-17-18-19-20-21
2) 1-2-3-8-9-10-11-12-13-14-16-17-18-19-20-21
3) 1-2-3-4-5-6-7-9-10-15-16-17-18-19-20-21

Berdasarkan 3 jalur tersebut didapatkan *test case* yang dapat dilihat pada Tabel 7.5 dan 7.6.

a. Deskripsi Pengujian

Tabel 7.5 Deskripsi pengujian menambah data dokumentasi pekerjaan

Nama proyek	Sistem Informasi Anggaran Pegawai
ID pengujian	BP-2
Fungsi	Fungsi menambah data dokumentasi pekerjaan
Deskripsi	*Test case* ini digunakan menguji fungsi tambah data dokumentasi pekerjaan pada bagian penambahan photo pekerjaan.
Perancang	Peneliti
Penguji	Penelti
Persyaratan Uji	Terhubung ke basis data dokumentasi

b. Prosedur pengujian

Tabel 7.6 Prosedur pengujian menambah data dokumentasi pekerjaan

No	Skenario pengujian	Data masukan	Hasil diharapkan	Hasil pengujian	Status
1	1. Penguji memilih menu tambah dokumentasi 2. Penguji mengisi *form* penambahan data dokumentasi kecuali photo sebelum dan sesudah 3. Penguji memilih tambah data	sebelum == null dan sesudah == null	Dokumentasi pekerjaan ditambahkan dengan 2 gambar *default*	Dokumentasi pekerjaan ditambahkan dengan 2 gambar *default*	Valid
2	1. Penguji memilih menu tambah dokumentasi 2. Penguji mengisi *form* penambahan data dokumentasi kecuali poto sebelum 3. Penguji memilih tambah data	sebelum == null dan sesudah == not null	Dokumentasi pekerjaan ditambahkan dengan photo sebelum berisi gambar *default*	Dokumentasi pekerjaan ditambahkan dengan photo sebelum berisi gambar *default*	Valid
3	1. Penguji memilih menu tambah dokumentasi 2. Penguji mengisi *form* penambahan data dokumentasi kecuali poto sesudah 3. Penguji memilih tambah data	sebelum == not null dan sesudah == null	Dokumentasi pekerjaan ditambahkan dengan photo sesudah berisi gambar *default*	Dokumentasi pekerjaan ditambahkan dengan photo sesudah berisi gambar *default*	Valid

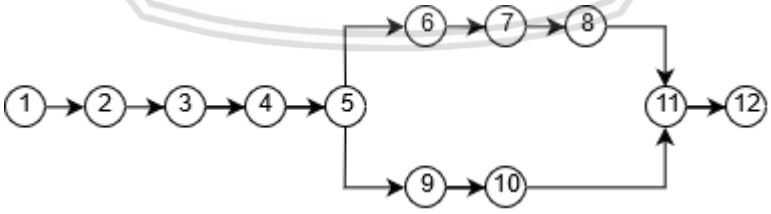
7.1.3 Pengujian *Basis Path* Menampilkan Data Keuangan

Pseducode dari hasil implementasi algoritme menampilkan data keuangan dapat ditunjukan pada Tabel 7.7. *Pseducode* menampilkan data keuangan akan digunakan sebagai acuan dalam penentuan *cyclomatic complexity* dan *independent path* sebagai sarana pengujian *basis path.*

Tabel 7.7 *Pseducode* menampilkan data keuangan

No	Pseducode
1	Mulai ----------1
2	Set id pekerjaan = ambil id pekerjaan dari *form* ----------2
3	Set nama pekerjaan = jalankan fungsi mengambil nama pekerjaan ----------3
4	Set nilai pekerjaan = jalankan fungsi mengambil nilai pekerjaan ----------4
5	Jika nilai kurang dari 1 ----------5
6	Set rumus = jalankan fungsi mengambil semua nilai pekerjaan ----------6
7	Set pesan notifikasi pada session ----------7
8	Beralih ke tampilan mengatur rumus nilai pekerjaan ----------8
9	Lainya
10	Jalankan fungsi mengambil data perhitungan keuangan ----------9
11	Beralih ke antarmuka untuk menampilkan data hasil perhitungan keuangan ----------10
12	Akhiri pengkondisian ----------11
13	Selesai ----------12

Berdasarkan *pseducode* di atas dapat dihasilkan *flow graph* sebagaimana ditunjukan pada gambar 7.4.



Gambar 7.4 *Flow graph* menampilkan data keuangan

Berdasarkan *flow graph* yang telah dibuat dapat ditentukan nilai *cyclomatic complexity* melalui perhitungan sebagai berikut :

V(G) = Jumlah busur-jumlah simpul+2 = 12-12+2 = 2
V(G) = Jumlah predicate node +1 = 1+1 = 2
V(G) = Jumlah region = 2

Hasil perhitungan *cyclomatic complexity* dapat digunakan sebagai acuan menentukan dua buah jalur *independent.* Ke dua jalur tersebut merupakan :

1) 1-2-3-4-5-6-7-8-11-12
2) 1-2-3-4-5-9-10-11-12

Mengacu pada ke dua jalur tersebut dapat ditentukan *test case* sebagaimana dijelaskan pada Tabel 7.8 dan Tabel 7.9.

a. Deskripsi Pengujian

Tabel 7.8 Deskripsi pengujian menampilkan data keuangan

Nama proyek	Sistem Informasi Anggaran Pegawai
ID pengujian	BP-3
Fungsi	Fungsi menampilkan data keuangan
Deskripsi	*Test case* ini digunakan untuk menguji fungsi menampilkan data keuangan untuk menghindari sistem menampilkan data perhitungan yang kosong.
Perancang	Peneliti
Penguji	Penelti
Persyaratan Uji	Terhubung ke basis data pegawai, kehadiran, penambahan, pengurangan, pekerjaan, dan rumus.

b. Deskripsi Pengujian

Tabel 7.9 Prosedur pengujian menampilkan data keuangan

No	Skenario pengujian	Data masukan	Hasil diharapkan	Hasil pengujian	Status
1	1. Penguji memilih menu lihat keuangan 2. Penguji memilih pekerjaan 3. Penguji memilih tombol lihat keuangan	nilai pekerjaan = 0	Tidak menampilkan hasil perhitungan keuangan dan beralih ke antarmuka merubah rumus nilai pekerjaan	Tidak menampilkan hasil perhitungan keuangan dan beralih ke antarmuka merubah rumus nilai pekerjaan	Valid
2	1. Penguji memilih menu lihat keuangan 2. Penguji memilih pekerjaan	nilai pekerjaan >= 1	Hasil perhitungan keuangan ditampilkan	Hasil perhitungan keuangan ditampilkan	Valid

	3. Penguji memilih tombol lihat keuangan				

7.1.4 Kesimpulan Pengujian *Basis Path*

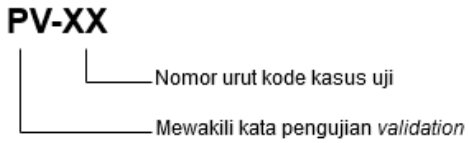
Dari hasil pengujian *basis path* yang telah dilakukan dapat di tarik kesimpulan sebagai berikut :

1. Perhitungan *cyclomatic complexity* pada ketiga fungsi mendapatkan nilai kurang dari 10. Berdasarkan teori McCabe (1996) dapat ditarik kesimpulan, tingkat kompleksitas ketiga fungsi tersebut masih di batas yang bisa diterima sehingga tidak diperlukan untuk melakukan pemecahan fungsi.
2. Pada pengujian fungsi mengalokasi pegawai setelah dilakukan perhitungan *cyclomatic complexity* pada *flow* graph menghasilkan tiga buah *independent path*. Hasil uji setiap *path* menggunakan *test case* memberikan status valid.
3. Pada pengujian fungsi menambah data dokumentasi pekerjaan setelah dilakukan perhitungan *cyclomatic complexity* pada *flow* graph menghasilkan tiga buah *independent path*. Hasil uji setiap *path* menggunakan *test case* menghasilkan status valid.
4. Hasil perhitungan *cyclomatic complexity* pada pengujian menampilkan data keuangan menghasilkan dua buah *independent path*, dimana hasil uji setiap *path* menggunakan *test case* menunjukan status valid.

7.2 Pengujian Validasi

Untuk mengatasi permasalahan waktu pada pengujian *white box*. Pressman (2010) menyebutkan terdapat metode pengujian *black box* sebagai jalan lain dan pelengkap dalam melakukan pengujian sistem. Pengujian validasi merupakan salah satu teknik yang terdapat pada metode pengujian *black box*. Menurut Pressman (2010) pengujian validasi berbasis skenario pada *use case* dapat mengungkap kesalahan interaksi pada sistem, sehingga sistem akan melakukan apa yang memang diinginkan oleh pengguna.

Pengujian validasi dilakukan untuk mengetahui bahwa sistem dapat berjalan sesuai harapan pengguna. Subbab ini berisi hasil pengujian validasi terhadap beberapa fungsi sistem informasi yang digunakan oleh pengguna untuk mengelola pegawai, mengelola dokumentasi, dan mengelola keuangan.



Gambar 7.5 Keterangan kodifikasi pengujian validasi

Pengujian validasi dilakukan berdasarkan skenario uji yang diperoleh dari beberapa alur *use case* mengelola pegawai, mengelola dokumentasi, dan

mengelola keuangan. Selanjutnya, kasus uji dapat diidentifikasi berdasarkan skenario uji yang telah ditentukan dan pengujian validasi dapat dilakukan. Kasus uji pengujian validasi memiliki kodifikasi seabagai identitas yang dapat ditunjukan pada Gambar 7.5.

7.2.1 Pengujian Validasi Mengelola Pegawai

Pada pengujian validasi mengelola pegawai akan menjelaskan pengujian pada beberapa fungsi yang dapat digunakan untuk mengelola data pegawai. Pada Tabel 7.10 menunjukan beberapa skenario yang mengacu pada alur *use case* Mengelola pegawai yang akan digunakan sebagai bahan identifikasi kasus uji.

Tabel 7.10 Matriks skenario pengujian validasi mengelola pegawai

No. Skenario	Alur Awal	Alur Alternatif
Skenario 1	*Basic Flow*	-
Skenario 2	*Basic Flow*	A1. Menambah data pegawai
Skenario 3	*Basic Flow*	A2. Mengubah data pegawai
Skenario 4	*Basic Flow*	A3. Menghapus data pegawai
Skenario 5	*Basic Flow*	A5. Mencetak data pegawai

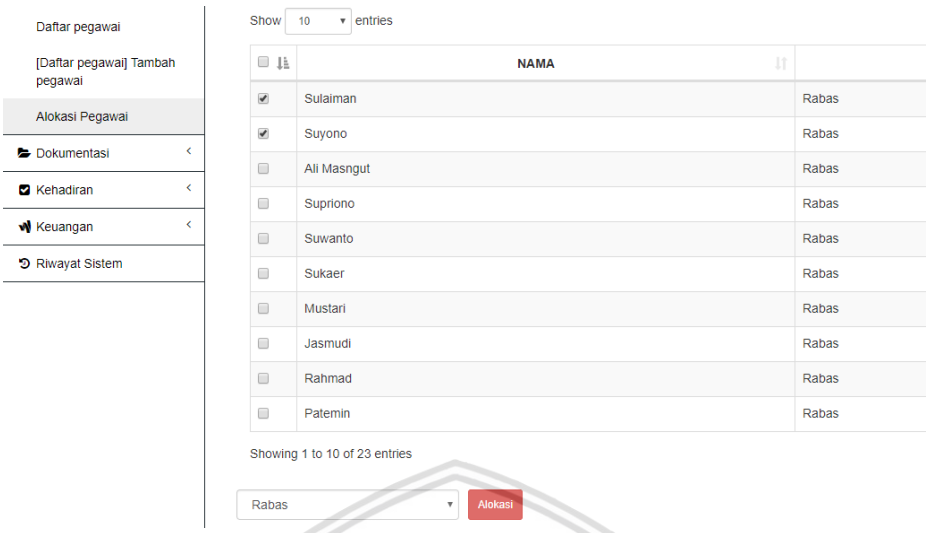
7.2.1.1 Pengujian Validasi Alokasi Pegawai

Pada Tabel 7.11 menunjukan deskripsi pengujian validasi dengan kode PV-1. Kasus uji PV-1 dilakukan untuk memastikan bahwa sistem dapat mengalokasi pegawai ke dalam pekerjaan dengan kode persyaratan KF-S-3.5. Kasus uji diidentifikasi berdasarkan skenario 1 pada Tabel 7.10.

a. Deskripsi pengujian

Tabel 7.11 Deskripsi pengujian alokasi pegawai

Nama Proyek	Sistem Informasi Anggaran Pegawai
Kode Pengujian	PV-1
Nomor Skenario	Skenario 1
Fungsi	Fungsi alokasi pegawai
Deskripsi	*Test case* ini digunakan untuk menguji fungsionalitas dengan kode KF-S-3.5. Dengan cara memilih dua pegawai untuk dialokasikan ke pekerjaan yang sudah tercatat di dalam sistem.
Perancang	Peneliti
Penguji	Peneliti
Persyaratan	Terhubung ke basis data pegawai



Gambar 7.6 Tampilan pengujian alokasi pegawai

b. Prosedur pengujian

Tabel 7.12 Prosedur pengujian alokasi pegawai

No	Prosedur Uji	Aktor	Data uji dimasukan	Hasil yang diharapkan	Hasil pengujian	Status
1	1. Penguji memilih menu alokasi pegawai 2. Penguji memilih nama pegawai 3. Penguji memilih pekerjaan 4. Penguji menekan tombol alokasi	Peneliti	2 nama pegawai dialokasi ke pekerjaan = Rabas	Jenis pekerjaan dua pegawai terpilih berubah menjadi = Rabas	Jenis pekerjaan dua pegawai terpilih berubah menjadi = Rabas	Valid

7.2.1.2 Pengujian Validasi Menambah Data Pegawai

Pada bagian Tabel 7.13 dapat dijelaskan deskripsi pengujian validasi fungsi menambah data pegawai dengan kode persyaratan KF-S-3.1. Pengujian dilakukan untuk memastikan bahwa sistem dapat digunakan untuk menambah data pegawai baru melalui sistem informasi yang telah dikembangkan. Deskripsi pengujian dungsi menambah data pegawai diidentifikasi mengacu pada skenario 2 pada Tabel 7.10

a. Deskripsi pengujian

Tabel 7.13 Deskripsi pengujian menambah data pegawai

Nama Proyek	Sistem Informasi Anggaran Pegawai
Kode Pengujian	PV-2

Nomor Skenario	Skenario 2
Fungsi	Fungsi menambah data pegawai
Deskripsi	*Test case* ini digunakan untuk menguji fungsionalitas dengan kode KF-S-3.1. Dengan cara memasukan data baru sesuai format yang ditentukan pada *form* tambah data pegawai.
Perancang	Peneliti
Penguji	Peneliti
Persyaratan	Terhubung ke basis data pegawai

Gambar 7.7 Tampilan pengujian menambah data pegawai

b. Prosedur pengujian

Tabel 7.14 Prosedur pengujian menambah data pegawai

No	Prosedur Uji	Aktor	Data uji dimasukan	Hasil yang diharapkan	Hasil pengujian	Status
1	1. Penguji memilih menu tambah pegawai 2. Penguji mengisi *form* penambahan data pegawai 3. Penguji memilih tombol tambah	Peneliti	Nama maksimal 100 karakter, memilih jenis kelamin, alamat maksimal 255 karakter, nomor telepon maksimal 20 karakter	Data pegawai masuk ke daftar pegawai	Data pegawai masuk ke daftar pegawai	Valid

7.2.1.3 Pengujian Validasi Mengubah Data Pegawai

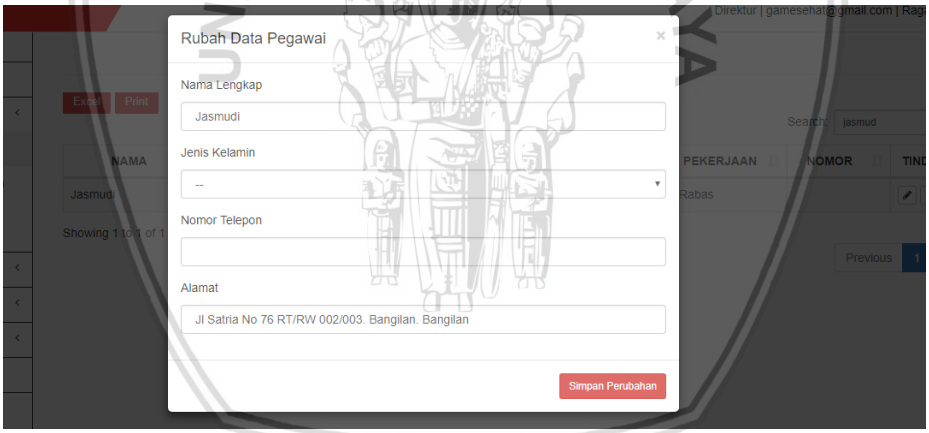
Pada Tabel 7.15 menjelaskan deskripsi pengujian validasi dengan kode pengujian PV-3. Pengujian validasi ini dilakukan untuk memastikan bahwa sistem

dapat mengubah data pegawai yang telah tersimpan sebelumnya pada sistem dengan kode persyaratan KF-S-3.2. Kasus uji PV-3 diidentifikasi berdasarkan skenario 3 pada Tabel 7.10.

a. Deskripsi pengujian

Tabel 7.15 Deskripsi pengujian mengubah data pegawai

Nama Proyek	Sistem Informasi Anggaran Pegawai
Kode Pengujian	PV-3
Nomor Skenario	Skenario 3
Fungsi	Fungsi mengubah data pegawai
Deskripsi	*Test case* ini digunakan untuk menguji fungsionalitas dengan kode KF-S-3.2. Dengan cara mengubah data pegawai menggunakan data baru untuk pegawai dengan nama = Jasmudi
Perancang	Peneliti
Penguji	Peneliti
Persyaratan	Terhubung ke basis data pegawai



Gambar 7.8 Tampilan pengujian mengubah data pegawai

b. Prosedur pengujian

Tabel 7.16 Prosedur pengujian mengubah data pegawai

No	Prosedur Uji	Aktor	Data uji dimasukan	Hasil yang diharapkan	Hasil pengujian	Status
1	1. Penguji memilih menu daftar pegawai 2. Penguji memilih menu untuk rubah pegawai	Peneliti	Rubah data pegawai dengan Nama = Jasmudi dengan persyaratan nama	Data pegawai baru tersimpan dan ditampilkan oleh sistem	Data pegawai baru tersimpan dan ditampilkan oleh sistem	Valid

	3. Penguji mengisi *form* dengan data baru pegawai 4. Penguji menekan tombol simpan perubahan		maksimal 100 karakter, memilih jenis kelamin, alamat maksimal 255 karakter, nomor telepon maksimal 20 karakter	pada daftar pegawai	pada daftar pegawai	

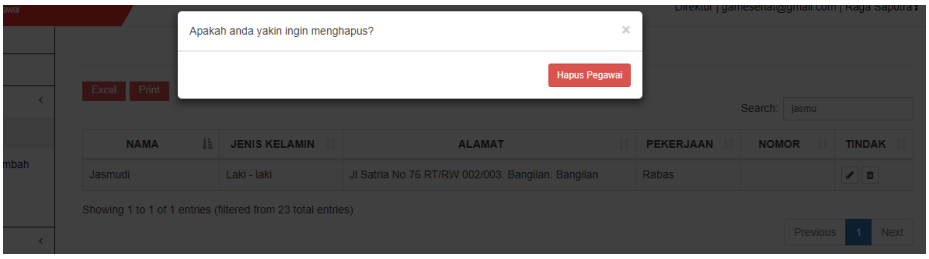
7.2.1.4 Pengujian Validasi Menghapus Data Pegawai

Deskripsi pengujian validasi untuk fungsi menghapus data pegawai terdapat pada Tabel 7.17. Pengujian dengan kode PV-4 dilakukan untuk memastikan bahwa sistem yang telah dibangun dapat menghapus data pegawai yang telah disimpan pada sistem dengan kode persyaratan KF-S-3.3. Deskripsi pengujian validasi pada fungsi ini diidentifikasi berdasarkan skenario 4 pada Tabel 7.10.

a. Deskripsi pengujian

Tabel 7.17 Deskripsi pengujian menghapus data pegawai

Nama Proyek	Sistem Informasi Anggaran Pegawai
Kode Pengujian	PV-4
Nomor Skenario	Skenario 4
Fungsi	Fungsi mengubah data pegawai
Deskripsi	*Test case* ini digunakan untuk menguji fungsionalitas dengan kode KF-S-3.3. Dengan cara menghapus data pegawai untuk pegawai dengan nama = Jasmudi
Perancang	Peneliti
Penguji	Peneliti
Persyaratan	Terhubung ke basis data pegawai



Gambar 7.9 Tampilan pengujian menghapus data pegawai

b. Prosedur pengujian

Tabel 7.18 Prosedur pengujian menghapus data pegawai

No	Prosedur Uji	Aktor	Data uji dimasukan	Hasil yang diharapkan	Hasil pengujian	Status
1	1. Penguji memilih menu daftar pegawai 2. Penguji memilih menu hapus pada salah satu pegawai 3. Penguji mengkonfirmasi penghapusan	Peneliti	Hapus data pegawai dengan Nama = Jasmudi	Data pegawai dengan nama Jasmudi terhapus dari daftar pegawai	Data pegawai dengan nama Jasmudi terhapus dari daftar pegawai	Valid

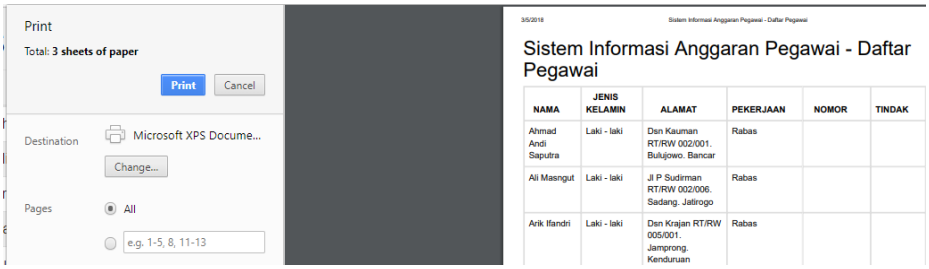
7.2.1.5 Pengujian Validasi Mencetak Data Pegawai

Pada Tabel 7.19 menampilkan informasi deskripsi pengujian fungsi mencetak data pegawai dengan kode persyaratan KF-S-3.4. Pengujian dengan kode PV-5 ini dilakukan untuk memastikan bahwa sistem yang telah dibangun dapat digunakan untuk mencetak data pegawai. Deskripsi pengujian PV-5 diidentifikasi berdasarkan skenario 5 pada Tabel 7.10.

a. Deskripsi pengujian

Tabel 7.19 Deskripsi pengujian mencetak data pegawai

Nama Proyek	Sistem Informasi Anggaran Pegawai
Kode Pengujian	PV-5
Nomor Skenario	Skenario 5
Fungsi	Fungsi mengubah data pegawai
Deskripsi	*Test case* ini digunakan untuk menguji fungsionalitas kode KF-S-3.4. Dengan cara mencetak seluruh pegawai untuk pekerjaan = rabas.
Perancang	Peneliti
Penguji	Peneliti
Persyaratan	Terhubung ke basis data pegawai



Gambar 7.10 Tampilan pengujian mencetak data pegawai

b. Prosedur pengujian

Tabel 7.20 Prosedur pengujian mencetak data pegawai

No	Prosedur Uji	Aktor	Data uji dimasukan	Hasil yang diharapkan	Hasil pengujian	Status
1	1. Penguji memilih menu daftar pegawai 2. Penguji memilih me-*filter* data pegawai yang akan dicetak (opsional) 3. Penguji memilih tombol untuk mencetak	Peneliti	Mencetak seluruh data pegawai dengan pekerjaan = rabas	Data pegawai tercetak	Data pegawai tercetak	Valid

7.2.2 Pengujian Validasi Mengelola Dokumentasi

Bagian ini menjelaskan pengujian validasi pada beberapa fungsi yang dapat digunakan oleh pengguna untuk mengelola dokumentasi pekerjaan melalui sistem yang telah dikembangkan. Pada Tabel 7.21 menunjukan beberapa skenario yang bersumber dari alur *use case* Mengelola Dokumentasi yang akan digunakan untuk mengidentifikasi kasus uji.

Tabel 7.21 Matriks skenario pengujian validasi mengelola dokumentasi

No. Skenario	Alur Awal	Alur Alternatif
Skenario 1	*Basic Flow*	-
Skenario 2	*Basic Flow*	A1. Mengubah data dokumentasi
Skenario 3	*Basic Flow*	A2. Menghapus data dokumentasi
Skenario 4	*Basic Flow*	A3. Mencetak data dokumentasi

7.2.2.1 Pengujian Validasi Menambah Data Dokumentasi

Pada Tabel 7.22 berisi deskripsi pengujian validasi dengan kode PV-6. Kasus uji PV-6 dilakukan untuk memastikan bahwa sistem dapat menambah data dokumentasi melalui fungsi menambah dokumentasi dengan kode persyaratan KF-S-4.1. Kasus uji diidentifikasi berdasarkan skenario 1 pada Tabel 7.21.

c. Deskripsi pengujian

Tabel 7.22 Deskripsi pengujian menambah data dokumentasi

Nama Proyek	Sistem Informasi Anggaran Pegawai
Kode Pengujian	PV-6
Nomor Skenario	Skenario 1
Fungsi	Fungsi menambah dokumentasi
Deskripsi	*Test case* ini digunakan untuk menguji fungsionalitas dengan kode KF-S-4.1. Dengan cara memasukan data baru sesuai format yang ditentukan pada *form* tambah dokumentasi untuk pekerjaan rabas.
Perancang	Peneliti
Penguji	Peneliti
Persyaratan	Terhubung ke basis data dokumentasi

Gambar 7.11 Tampilan pengujian menambah data dokumentasi

d. Prosedur pengujian

Tabel 7.23 Prosedur pengujian menambah data dokumentasi

No	Prosedur Uji	Aktor	Data uji dimasukan	Hasil yang diharapkan	Hasil pengujian	Status
1	1. Penguji memilih menu tambah dokumentasi 2. Penguji mengisi *form*	Peneliti	Masukan dokumentasi pekerjaan = Rabas	Data dokumentasi masuk ke daftar dokumentasi	Data dokumentasi masuk ke daftar dokumentasi	Valid

	penambahan data dokumentasi 3. Penguji memilih tambah data					

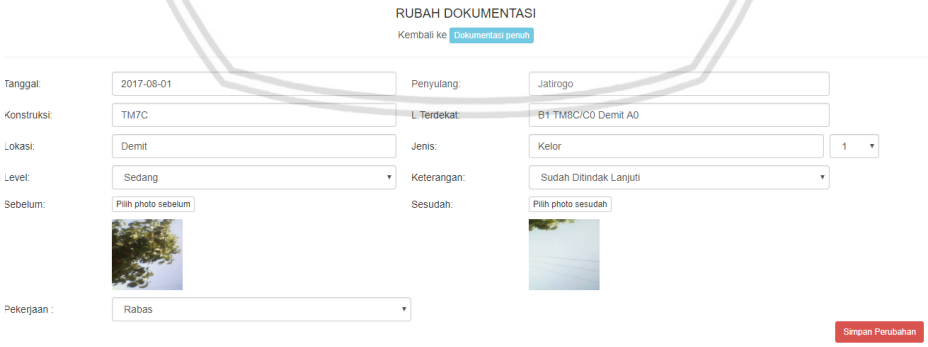
7.2.2.2 Pengujian Validasi Mengubah Data Dokumentasi

Pada Tabel 7.24 menjelaskan deskripsi pengujian validasi dengan kode pengujian PV-7. Kasus uji PV-7 dilakukan untuk memastikan bahwa sistem dapat mengubah data dokumentasi melalui fungsi mengubah dokumentasi. Kasus uji diidentifikasi berdasarkan skenario 2 pada Tabel 7.21.

a. Deskripsi pengujian

Tabel 7.24 Deskripsi pengujian mengubah data dokumentasi

Nama Proyek	Sistem Informasi Anggaran Pegawai
Kode Pengujian	PV-7
Nomor Skenario	Skenario 2
Fungsi	Fungsi mengubah dokumentasi
Deskripsi	Test case ini digunakan untuk menguji fungsionalitas dengan kode KF-S-4.2. Dengan mengubah data menggunakan data baru untuk dokumentasi pekerjaan rabas dengan lokasi = sugihan.
Perancang	Peneliti
Penguji	Peneliti
Persyaratan	Terhubung ke basis data dokumentasi



Gambar 7.12 Tampilan pengujian mengubah data dokumentasi

b. Prosedur pengujian

Tabel 7.25 Prosedur pengujian mengubah dokumentasi

No	Prosedur Uji	Aktor	Data uji dimasukan	Hasil yang diharapkan	Hasil pengujian	Status

1	1. Penguji memilih menu daftar dokumentasi 2. Penguji memilih menu rubah dari salah satu dokumentasi 3. Penguji mengisi *form* dengan data baru 4. Penguji menekan tombol simpan	Peneliti	Rubah data dokumentasi rabas lokasi = sugihan	Data dokumentasi baru menggantikan yang lama pada daftar dokumentasi	Data dokumentasi baru menggantikan yang lama pada daftar dokumentasi	Valid

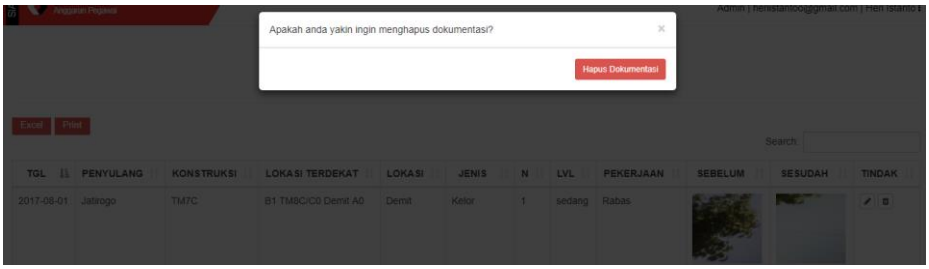
7.2.2.3 Pengujian Validasi Menghapus Data Dokumentasi

Pada Tabel 7.26 berisi informasi deskripsi pengujian fungsi menghapus data dokumentasi dengan kode persyaratan KF-S-4.3. Pengujian dilakukan untuk memastikan bahwa sistem dapat digunakan untuk menghapus data dokumentasi. Deskripsi pengujian fungsi menghapus data dokumentasi diidentifikasi berdasarkan skenario 3 pada Tabel 7.21.

a. Deskripsi pengujian

Tabel 7.26 Deskripsi pengujian menghapus data dokumentasi

Nama Proyek	Sistem Informasi Anggaran Pegawai
Kode Pengujian	PV-8
Nomor Skenario	Skenario 3
Fungsi	Fungsi menghapus dokumentasi
Deskripsi	*Test case* ini digunakan untuk menguji fungsionalitas dengan kode KF-S-4.3. Dengan menghapus satu data untuk dokumentasi rabas dengan lokasi = sugihan
Perancang	Peneliti
Penguji	Peneliti
Persyaratan	Terhubung ke basis data dokumentasi



Gambar 7.13 Tampilan pengujian menghapus data dokumentasi

b. **Prosedur pengujian**

Tabel 7.27 Prosedur pengujian menghapus data dokumentasi

No	Prosedur Uji	Aktor	Data uji dimasukan	Hasil yang diharapkan	Hasil pengujian	Status
1	1. Penguji memilih menu daftar dokumentasi 2. Penguji memilih menu hapus dari salah satu dokumentasi 3. Penguji mengkonfirmasi penghapusan	Peneliti	Hapus sebuah data dokumentasi rabas dengan lokasi = Sugihan	Data dokumentasi terpilih terhapus	Data dokumentasi terpilih terhapus	Valid

7.2.2.4 Pengujian Validasi Mencetak Data Dokumentasi

Pada Tabel 7.28 berisi informasi deskripsi pengujian fungsi mencetak dokumentasi dengan kode persyaratan KF-S-4.4. Pengujian dilakukan untuk memastikan bahwa sistem dapat digunakan untuk mencetak data dokumentasi pekerjaan. Deskripsi pengujian diidentifikasi berdasarkan skenario 4 pada Tabel 7.21.

a. **Deskripsi pengujian**

Tabel 7.28 Deskripsi pengujian mencetak data dokumentasi

Nama Proyek	Sistem Informasi Anggaran Pegawai
Kode Pengujian	PV-9
Nomor Skenario	Skenario 4
Fungsi	Fungsi mencetak data dokumentasi
Deskripsi	*Test case* ini digunakan untuk menguji fungsionalitas dengan kode KF-S-4.4. Dengan mencetak seluruh dokumentasi untuk pekerjaan = rabas, pada bulan januari
Perancang	Peneliti
Penguji	Peneliti
Persyaratan	Terhubung ke basis data dokumentasi

Gambar 7.14 Tampilan pengujian mencetak data dokumentasi

b. Prosedur pengujian

Tabel 7.29 Prosedur pengujian mencetak data dokumentasi

No	Prosedur Uji	Aktor	Data uji dimasukan	Hasil yang diharapkan	Hasil pengujian	Status
1	1. Penguji memilih menu daftar dokumentasi 2. Penguji me *filter* data (opsional) 3. Penguji memilih menu untuk mencetak	Peneliti	Mencetak seluruh dokumentasi pekerjaan rabas , bulan = 2018-01	Data dokumentasi tercetak	Data dokumentasi tercetak	Valid

7.2.3 Pengujian Validasi Mengelola Keuangan

Bagian ini menjelaskan pengujian validasi pada beberapa fungsi yang dapat digunakan oleh pengguna untuk mengelola data keuangan melalui sistem yang telah dikembangkan. Pada Tabel 7.30 menunjukan beberapa skenario yang bersumber dari alur *use-case* Mengelola Keuangan yang akan digunakan untuk menegidentifikasi kasus uji.

Tabel 7.30 Matriks skenario pengujian validasi mengelola keuangan

No. Skenario	Alur Awal	Alur Alternatif
Skenario 1	*Basic Flow*	-
Skenario 2	*Basic Flow*	A2. Menambah data keuangan
Skenario 3	*Basic Flow*	A4. Membatalkan data keuangan
Skenario 4	*Basic Flow*	A6. Mencetak data keuangan
Skenario 5	*Basic Flow*	A7. Mengatur rumus keuangan

7.2.3.1 Pengujian Validasi Menampilkan Data Keuangan

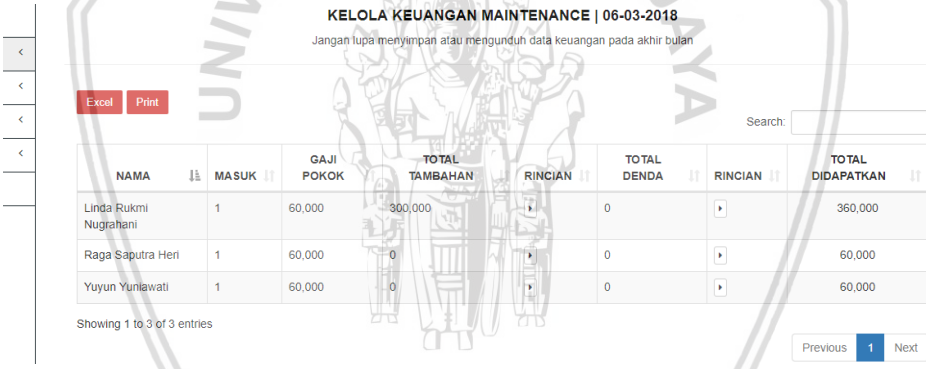
Pada Tabel 7.31 berisi deskripsi pengujian validasi fungsi menampilkan data keuangan dengan kode persyaratan KF-S-6-5. Pengujian dilakukan untuk memastikan bahwa sistem mampu menampilkan data keuangan setelah dilakukan

perhitungan secara otomatis. Deskripsi pengujian fungsi menampilkan data keuangan diidentifikasi berdasarakan skenario 1 pada Tabel 7.30.

a. Deskripsi pengujian

Tabel 7.31 Deskripsi pengujian menampilkan data keuangan

Nama Proyek	Sistem Informasi Anggaran Pegawai
Kode Pengujian	PV-10
Nomor Skenario	Skenario 1
Fungsi	Fungsi menampilkan data keuangan
Deskripsi	*Test case* ini digunakan untuk menguji fungsionalitas dengan kode KF-S-6.5. Dengan cara memilih menu lihat keuangan untuk pekerjaan rabas
Perancang	Peneliti
Penguji	Peneliti
Persyaratan	Terhubung ke basis data keuangan



Gambar 7.15 Tampilan pengujian menampilkan data keuangan

b. Prosedur pengujian

Tabel 7.32 Prosedur pengujian menampilkan data keuangan

No	Prosedur Uji	Aktor	Data uji dimasukan	Hasil yang diharapkan	Hasil pengujian	Status
1	1. Penguji memilih menu lihat keuangan 2. Penguji memilih pekerjaan rabas 3. Penguji menekan tombol lihat keuangan	Peneliti	Pekerjaan = rabas	Data keuangan pekerjaan terpilih dihitung dan ditampilkan	Data keuangan pekerjaan terpilih dihitung dan ditampilkan	Valid

7.2.3.2 Pengujian Validasi Menambah Data Keuangan

Pada Tabel 7.33 menjelaskan deskripsi pengujian dengan kode PV-11. Pengujian dengan kasus uji kode PV-11 dilakukan untuk memastikan bahwa sistem dapat digunakan untuk menambah data keuangan pegawai. Kasus uji diidentifikasi berdasarkan skenario 2 pada Tabel 7.30 dengan persyaratan fungisonal nomor KF-S-6.1.

a. Deskripsi pengujian

Tabel 7.33 Deskripsi pengujian menambah data keuangan

Nama Proyek	Sistem Informasi Anggaran Pegawai
Kode Pengujian	PV-11
Nomor Skenario	Skenario 2
Fungsi	Fungsi menambah data keuangan
Deskripsi	*Test case* ini digunakan untuk menguji fungsionalitas sistem dengan kode KF-S-6.1. Dengan menambah keuangan untuk pegawai dengan nama = 'Sulaiman'
Perancang	Peneliti
Penguji	Peneliti
Persyaratan	Terhubung ke basis data keuangan

Gambar 7.16 Tampilan pengujian menambah data keuangan

b. Prosedur pengujian

Tabel 7.34 Prosedur pengujian menambah data keuangan

No	Prosedur Uji	Aktor	Data uji dimasukan	Hasil yang diharapkan	Hasil pengujian	Status
1	1. Penguji memilih menu rincian penambahan keuangan pegawai dengan nama = sulaiman	Peneliti	Menambah data tunjangan untuk pegawai = 'Sulaiman', dengan nominal = 50000, dan	Data penambahan keuangan pegawai dengan nama = sulaiman bertambah	Data penambahan keuangan pegawai dengan nama = sulaiman bertambah	Valid

	pada halaman lihat keuangan pekerjaan rabas 2. Penguji memilih menu tambah keuangan 3. Penguji mengisi *form* penambahan keuangan 4. Penguji memilih tambah data		perihal dengan maksimal 100 karakter			

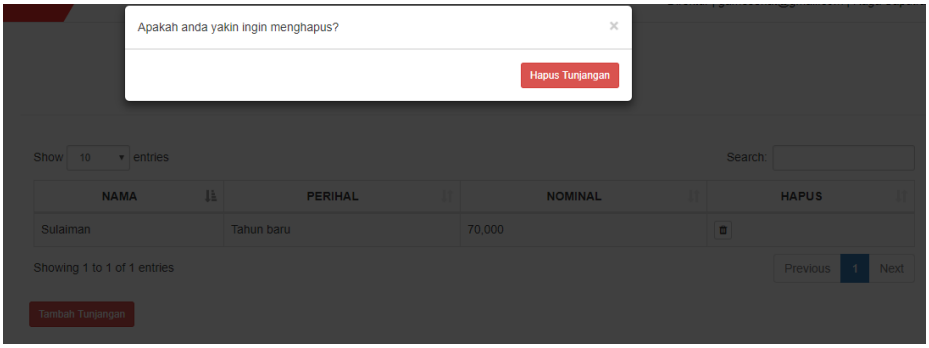
7.2.3.3 Pengujian Validasi Membatalkan Data Keuangan

Pada Tabel 7.35 berisi deskripsi pengujian validasi dengan kode PV-12. Kasus uji PV-12 dilakukan untuk memastikan bahwa sistem yang telah dikembangkan dapat digunakan untuk membatalkan atau menghapus data keuangan berdasarkan kode persyaratan KF-S-6.2. Kasus uji diidentifikasi berdasarkan skenario 3 pada Tabel 7.30.

a. Deskripsi pengujian

Tabel 7.35 Deskripsi pengujian membatalkan data keuangan

Nama Proyek	Sistem Informasi Anggaran Pegawai
Kode Pengujian	PV-12
Nomor Skenario	Skenario 3
Fungsi	Fungsi membatalkan data keuangan
Deskripsi	*Test case* ini digunakan untuk menguji fungsionalitas kode KF-S-6.2 Dengan membatalkan salah satu penambahan keuangan untuk pegawai dengan nama = 'Sulaiman'
Perancang	Peneliti
Penguji	Peneliti
Persyaratan	Terhubung ke basis data keuangan



Gambar 7.17 Tampilan pengujian membatalkan data keuangan

b. Prosedur pengujian

Tabel 7.36 Prosedur pengujian membatalkan data keuangan

No	Prosedur Uji	Aktor	Data uji dimasukan	Hasil yang diharapkan	Hasil pengujian	Status
1	1. Penguji memilih menu rincian penambahan keuangan pegawai dengan nama = sulaiman pada halaman lihat keuangan pekerjaan rabas 2. Penguji memilih tombol hapus rincian dari salah satu rincian penambahan keuangan 3. Penguji mengkonfirmasi penghapusan penambahan keuangan	Peneliti	Membatalkan salah satu data penambahan untuk pegawai dengan nama = 'Sulaiman'	Data penambahan keuangan pegawai dengan nama sulaiman berkurang	Data penambahan keuangan pegawai dengan nama sulaiman berkurang	Valid

7.2.3.4 Pengujian Validasi Mencetak Data Keuangan

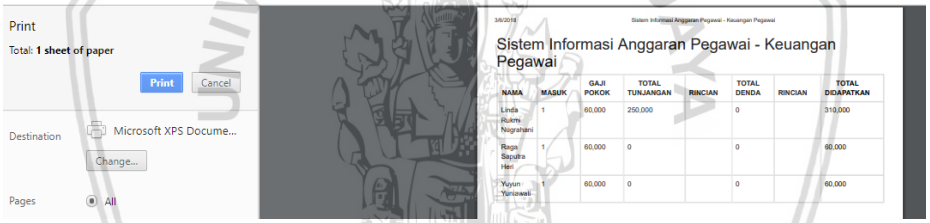
Pada Tabel 7.37 terdapat informasi deskripsi pengujian validasi fungsi mencetak data keuangan dengan kode persyaratan KF-S-6.3. Pengujian dengan kode PV-13 ini dilakukan untuk memastikan sistem yang telah dikembangkan dapat digunakan

untuk mencetak data hasil perhitungan keuangan. Deskripsi pengujian fungsi mencetak data keuangan diidentifikasi berdasarkan skenario 4 pada Tabel 7.30.

a. Deskripsi pengujian

Tabel 7.37 Deskripsi pengujian mencetak data keuangan

Nama Proyek	Sistem Informasi Anggaran Pegawai
Kode Pengujian	PV-13
Nomor Skenario	Skenario 4
Fungsi	Fungsi mencetak data keuangan
Deskripsi	*Test case* ini digunakan untuk menguji kode fungsionalitas KF-S-6.3 Dengan mencetak data keuangan dari seluruh pegawai pada pekerjaan rabas
Perancang	Peneliti
Penguji	Peneliti
Persyaratan	Terhubung ke basis data keuangan



Gambar 7.18 Tampilan pengujian mencetak data keuangan

b. Prosedur pengujian

Tabel 7.38 Prosedur pengujian mencetak data keuangan

No	Prosedur Uji	Aktor	Data uji dimasukan	Hasil yang diharapkan	Hasil pengujian	Status
1	1. Penguji memilih menu lihat keuangan pekerjaan rabas 2. Penguji memilih menu *print*	Peneliti	Mencetak seluruh keuangan pegawai pada pekerjaan = Rabas	Keuangan pegawai pekerjaan rabas tercetak	Keuangan pegawai pekerjaan rabas tercetak	Valid

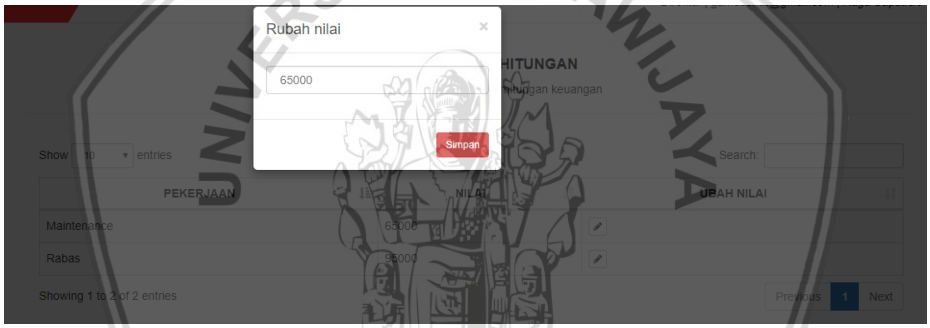
7.2.3.5 Pengujian Validasi Mengatur Rumus Keuangan

Pada Tabel 7.39 berisi deskripsi pengujian validasi dengan kode PV-14. Kasus uji dengan persyaratan KF-S-6.4 ini dilakukan untuk memastikan bahwa sistem dapat digunakan untuk mengatur rumus perhitungan keuangan. Kasus uji diidentifikasi berdasarkan skenario 5 pada Tabel 7.30.

a. **Deskripsi pengujian**

Tabel 7.39 Deskripsi pengujian mengatur rumus keuangan

Nama Proyek	Sistem Informasi Anggaran Pegawai
Kode Pengujian	PV-14
Nomor Skenario	Skenario 5
Fungsi	Fungsi mengatur rumus keuangan
Deskripsi	*Test case* ini digunakan menguji fungsionalitas kode KF-S-6.4 Dengan mengubah rumus perhitungan pekerjaan = Rabas menjadi 65.000
Perancang	Peneliti
Penguji	Peneliti
Persyaratan	Terhubung ke basis data keuangan



Gambar 7.19 Tampilan pengujian mengatur rumus keuangan

b. **Prosedur pengujian**

Tabel 7.40 Prosedur pengujian mengatur rumus keuangan

No	Prosedur Uji	Aktor	Data uji dimasukan	Hasil yang diharapkan	Hasil pengujian	Status
1	1. Penguji memilih menu atur rumus 2. Penguji menekan tombol untuk mengubah rumus 3. Penguji memasukan nilai rumus = 65.000 4. Penguji memilih tombol simpan	Peneliti	Rubah rumus perhitungan pekerjaan rabas rabas menjadi 65.000	Nilai rumus pekerjaan rabas berubah menjadi 65.000	Nilai rumus pekerjaan rabas berubah menjadi 65.000	Valid

7.2.4 Kesimpulan Pengujian Validasi

Berdasarkan pengujian validasi yang telah dilakukan pada skenario yang dibuat berdasarkan alur *use case* mengelola pegawai, mengelola dokumentasi, dan mengelola keuangan didapatkan hasil valid untuk semua skenario yang diujikan.

7.3 Pengujian Kompatibilitas

Pengujian kompatibilitas dilakukan menggunakan perangkat lunak bantuan *SortSite* versi 5.27.804.0. Dari pengujian ini dihasilkan pengetahuan versi peramban web apa saja yang mendukung berjalanya sistem informasi dengan baik. Pada Tabel 7.41 dapat dilihat jenis dan versi peramban web yang digunakan dalam pengujian kompatibilitas.

Tabel 7.41 Jenis dan versi peramban web

No	Nama Peramban web	Versi Peramban web
1	Internet Explorer (IE)	9, 10, 11
2	Microsoft Edget (Edge)	15
3	Firefox	56
4	Safari	<= 10, 11
5	Opera	47
6	Chrome	61
7	iOS	<= 9, 10, 11
8	Android	<= 3, 4
9	BlackBerry	<= 7.1, 10.0

Gambar 7.20 Hasil pengujian kompatibilitas

Terdapat 3 kategori masalah yang dapat diketahui dari hasil pengujian kompatibilitas menggunakan *SortSite* yaitu, *crictical issues* (merah), *major issues*, dan *minor issues*. Penjelasan dari ketiga kategori bisa dilihat pada Tabel 7.42.

Tabel 7.42 Kategori masalah kompatibilitas

	Tidak terdapat permasalahan pada konten atau fungsionalitas.
	Konten atau fungsionalitas hilang.
	Masalah mayor pada *layout* atau performa
	Masalah minor pada *layout* atau performa

Dari hasil pengujian kompatibilitas yang terdapat pada Gambar 7.20 bisa diambil kesimpulan bahwa sistem yang telah di kembangkan mampu berjalan dengan baik di sembilan peramban web yang diujikan. Namun masih ditemukan beberapa masalah minor pada pengujian di mana pada beberapa peramban web penulisan css dan penggunaan versi css belum didukung pada peramban tersebut.

BAB 8 PENUTUP

8.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, dapat diambil kesimpulan sebagai berikut:

1. Hasil analisis persyaratan sistem merupakan dokumentasi yang memuat informasi hasil identifikasi tipe pemangku kepentingan, masalah yang dihadapi oleh pemangku kepentingan, dan identifikasi kebutuhan pemangku kepentingan dan pengguna. Selanjutnya kegiatan analisis terhadap hasil identifikasi kebutuhan pemangku kepentingan dan pengguna dilakukan sehingga persyaratan sistem dapat teridentifikasi. Persyaratan sistem memuat informasi dari 8 fitur, 25 persyaratan fungsional, dan 1 persyaratan nonfungsional dari sistem informasi yang mendemonstrasikan fungsi-fungsi untuk mendukung proses penggajian pegawai dan dokumentasi pekerjaan. Hasil analisis persyaratan sistem juga berisi informasi kemampuan sistem yang didokumentasikan ke dalam 8 *use case*, dan urutan aktivitas yang dilakukan oleh pengguna dan sistem yang divisualisaikan ke dalam *activity diagram*.
2. Perancangan sistem informasi yang telah dilakukan berdasarkan hasil analisis persyaratan menghasilkan model interaksi objek yang divisualisasikan ke dalam *sequence diagram,* model objek yang divisualisasikan ke dalam *class diagram*, PDM yang memvisualisasikan 10 tabel sebagai media penyimpanan data sistem informasi, beberapa rancangan algoritme sistem, dan beberapa rancangan antarmuka pengguna.
3. Sistem informasi yang mendemonstrasikan fungsi-fungsi untuk mendukung proses penggajian pegawai dan dokumentasi pekerjaan telah diimplementasikan berdasarkan hasil perancangan sistem menggunakan kerangka kerja Laravel. Hasil implementasi sistem informasi ini memberikan fitur identifikasi identitas pengguna, pengelolaan data pekerjaan, pengelolaan data pegawai, pengelolaan data dokumentasi, pengelolaan data kehadiran, pengelolaan data keuangan, pencatatan dan penyajian aktivitas pengguna sistem, serta pengelolaan data petugas admin.
4. Hasil pengujian sistem bisa dibedakan menjadi 2, yaitu:
    a. Hasil pengujian persyaratan fungsional menggunakan metode pengujian *basis path* dan pengujian validasi yang dilakukan pada beberapa alur *use case* mengelola pegawai, mengelola dokumentasi, dan mengelola keuangan, semuanya dinyatakan valid. Ditunjukan dengan kesamaan antara hasil pengujian dengan hasil yang diharapkan.
    b. Pengujian persyaratan nonfungsional menggunakan metode pengujian kompatibilitas menghasilkan kesimpulan bahwa meski terdapat beberapa *minnor issues,* sistem mampu berjalan dengan baik di sembilan peramban web yang berbeda.

Dari kesimpulan yang telah disebutkan maka secara umum sebuah prototipe sistem informasi yang dapat mendemonstrasikan fungsi-fungsi untuk mendukung proses penggajian pegawai dan dokumentasi pekerjaan telah dapat dibangun.

8.2 Saran

1. Dapat dilakukan pengujian pada proses bisnis yang diusulkan oleh peneliti untuk mengetahui tingkat efesiensi proses bisnis baru setelah dijalankan pada organisasi, yang hasilnya dapat digunakan sebagai acuan dalam pengembangan sistem selanjutnya.
2. Dapat dilakukan evaluasi antarmuka pengguna untuk mengetahui tingkat *usability* dari sistem yang telah dibuat, dan hasilnya dapat digunakan sebagai acuan dalam memperbaiki antarmuka pada pengembangan selanjutnya.

DAFTAR PUSTAKA

Becker, J., Kugeler, M. & Rosemann, M., 2013. *Process Management: A Guide for the Design of Business Processes*. Berlin: Springer-Verlag.

Bittner, K. & Spence, I., 2002. *Use Case Modeling*. Boston: Addison Wesley.

Booch, G., Rumbaugh, J. & Jacobson, I., 1998. *The Unified Modelling Language User Guide*. Boston: Addison Wesley.

Herlambang, T. W., Aknuranda, I. & Saputra, M. C., 2017. Pengembangan Sistem Informasi Kesehatan Berbasis Web Berdasarkan Model Organisasi Dan Manajemen Kesehatan Primer "ANDAL". *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(4), pp. 1602-1610.

Krol, M. & Reich, D.L., 1999. *Object-Oriented Analysis and Design of a Health Care Management Information System*. *Journal of Medical System*, 24(2), pp.145-158

Laravel, 2014. *Laravel Philosophy.* [Online] Avaliable at: https://laravel.com/docs/4.2/introduction [Diakses 18 April 2018].

Leffingwell, D. & Widrig, D., 2003. *Managing Software Requirements: A Use Case Approach.* 2nd penyunt. Boston: Addison Wesley.

McCabe, T. J., Watson, A. H., 1996. *Structured Testing: A Testing Methodology Using the Cyclomatic Complexity Metric*. NIST Special Publication 500-235.

Monk, E. & Wagner, B., 2013. *Concepts in Enterprise Resource Planning. 4th penyunt*. Boston: Course Technology.

Object Management Group, 2005. *Unified Modelling Language: Superstructure.* [Online] Available at: http://www.omg.org/spec/UML/2.0/Superstructure/PDF/ [Diakses 19 Maret 2018].

Object Management Group, 2011. *Business Process Model and Notation (BPMN).* [Online] *Availble* at: http://www.omg.org/spec/BPMN/2.0/PDF [Diakses 19 Maret 2018].

Odeh, M., Haeur, T., McClatchey, R. & Solomonides, T., 2003. *A Use-Case Driven Approach in Requirements Engineering: The Mammogrid Project*. Coldharbour Lane, Centre for Complex Cooperative Systems.

Pressman, R. S., 2010. *Software Engineering: A Practitioner's Approach*. 7th penyunt. New York: McGraw-Hill.

Przybyłek, A., 2011. *Bridging the gap between business process models and use-case models*. Gdańsk, Gdańsk University of Technology.

Sommerville, I., 2011. *Software Engineering*. 9th penyunt. Boston: Addison-Wesley.